Calhoun
Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1990-09

Further calculations of the performance of turbofan engines incorporating a wave rotor

Roberts, James W.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/34929

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

DUDLEY KNOX LIBRARY

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A240 867

DTIC
ELECTE
SEP 27 1991
S B D

THESIS

FURTHER CALCULATIONS OF THE PERFORMANCE OF TURBOFAN ENGINES INCORPORATING A WAVE ROTOR

by

James W. Roberts

September 1990

Thesis Advisor: Raymond P. Shreeve

Approved for public release; distribution is unlimited

91-11620

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION; 8b OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)
FURTHER CALCULATIONS OF THE PERFORMANCE OF TURBOFAN ENGINES INCORPORATING A WAVE ROTOR

12 PERSONAL AUTHOR(S)
Roberts, James W.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1990, September	108

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Turbofan Engines; Turbofan Engines with a Wave Rotor

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Two recent computer programs, WRCOMP and ENGINE, by A. Mathur, were used to examine the performance to be gained by incorporating a wave rotor component in a turbofan engine with mixed exhausts. The programs were transferred to a VAX-2000 computer, extended, and test cases reported by A. Mathur were successfully reproduced. A comparison was made between ENGINE, in which real gas effects are accounted for, and ONX (by J. Mattingly) in which constant specific heats are used. The inclusion of real gas effects proved to have a significant impact on the predicted performance. An extension of Mathur's results was made by varying the overall pressure ratio in the wave-turbofan engine. Further cycle studies and experiments to measure wave rotor component performance are recommended.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION: Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL: Prof. Raymond P. Shreeve	22b TELEPHONE (Include Area Code): (408) 646-2593; 22c OFFICE SYMBOL: Code AA/Sf

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

U.S. Government Printing Office: 1986—606-24.

Approved for public release; distribution is unlimited

Further Calculations of the Performance of Turbofan Engines Incorporating a Wave Rotor

by

James W. Roberts
Lieutenant, United States Navy
B.S., United States Naval Academy, 1981

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1990

Author: [redacted]
James W. Roberts

Approved by: [redacted]
Raymond P. Shreeve, Thesis Advisor

[redacted]
Garth V. Hobson, Second Reader

[redacted]
E. Roberts Wood, Chairman
Department of Aeronautical Engineering

ABSTRACT

Two recent computer programs, WRCOMP and ENGINE, by A. Mathur, were used to examine the performance to be gained by incorporating a wave rotor component in a turbofan engine with mixed exhausts. The programs were transferred to a VAX-2000 computer, extended, and test cases reported by A. Mathur were successfully reproduced. A comparison was made between ENGINE, in which real gas effects are accounted for, and ONX (by J. Mattingly) in which constant specific heats are used. The inclusion of real gas effects proved to have a significant impact on the predicted performance. An extension of Mathur's results was made by varying the overall pressure ratio in the wave-turbofan engine. Further cycle studies and experiments to measure wave rotor component performance are recommended.

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

TABLE OF SYMBOLS

SUBSCRIPT	DEFINITION
LPC = f	Fan or low pressure compressor
LPT	Low pressure turbine
HPC	High pressure compressor
HPT	High pressure turbine
OPR	Overall pressure ratio

TEXT	DEFINITION
AMACH8	Low pressure turbine exit Mach number
AMEXIT	Exit Mach number for wave rotor component
β	Bypass ratio
CFL	Courant-Friedrichs-Levy condition
Cp_c	Compressor value for specific heat at constant pressure
Cp_t	Turbine value for specific heat at constant
f	Fuel-air ratio
F/m = ST	Specific thrust
h	Altitude (ft)
h	Enthalpy
M	Mach number
min	Mass flow rate in
mout	Mass flow rate out
π = TPR	Total Pressure Ratio

PRLC	Pressure ratio across tne low pressure compressor
PRHC	Pressure ratio across the high pressure compressor
°R	Degrees Rankine
RREF	Reference density inside the wave rotor
SFC	Specific fuel consumption
SPR	Static Pressure Ratio
τ = TTR	Total Temperature Ratio
T	Temperature
TIT	Turbine Inlet Temperature

ACKNOWLEDGMENTS

Thanks to my best friend and wife, Kim Roberts, for her support and encouragement, and to Ray Shreeve, for making this a challenging and rewarding experience.

The work is dedicated to our son, Adam James Roberts.

I. INTRODUCTION

Interest by DARPA (Defense Advance Research Projects Agency) and NAVAIRSYSCOM (Naval Air Systems Command) in obtaining increased range for unmanned air vehicles (UAV's) prompted the present study of wave rotor applications in engines. The wave rotor offers potential to achieve lower specific fuel consumption (SFC) and higher specific thrust by allowing increased cycle temperatures without increasing turbine inlet temperature. The goal of this study was to examine the potential benefits of incorporating a wave rotor in a turbofan engine with mixed exhausts. The results showed that significant benefits could be obtained in both SFC and specific thrust depending upon the selection of other cycle parameters.

Chapter II describes the wave rotor concept, presents a general background on trends in gas turbine technology, gives a description of a wave rotor engine component and introduces the two computer programs (ENGINE and ONX) which were used in the study. Chapter III gives a detailed description of the ENGINE code and presents a comparison of results obtained with ENGINE and ONX. Chapter IV gives a description of the WRCOMP program, which analyses the unsteady wave rotor flow, and its interface with the ENGINE program. Chapter V gives results obtained using the codes for the performance of a turbofan

with mixed exhausts, with and without a wave rotor component. Conclusions and recommendations for further work are given in Chapter VI. Details are given in the appendices. Appendices A and B contain procedural guides, sample results and program listings for ENGINE and WRCOMP, respectively, while Appendix C describes file transfer and graphics procedures which were used in the course of the study.

II. BACKGROUND

A. POTENTIAL FOR WAVE ROTOR APPLICATIONS IN ENGINES

A wave rotor is a partial admission device which causes one gas to compress another by wave propagation [Ref. 1:p. 1]. A series of straight compression tubes are aligned on a rotor encased in a drum. The rotor rotates on a fixed axis. Flow in and out of the wave rotor is steady while flow inside the wave rotor tubes is unsteady. Figure 1 shows an example of a wave rotor component. The single wave rotor component serves the same function as the combination of a compressor and a turbine mounted on the same shaft.

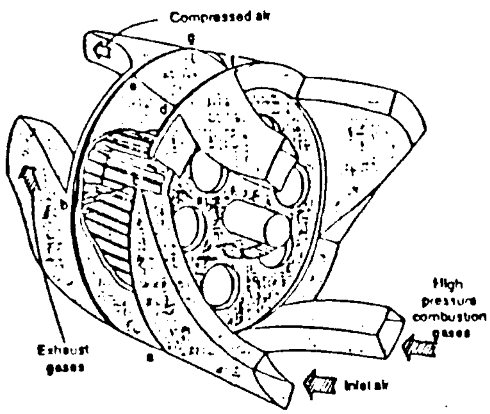


Figure 1. Wave Rotor Component [Ref. 1:p. 294]

Figure 2 shows a turbofan engine configured with an axially mounted wave rotor component.

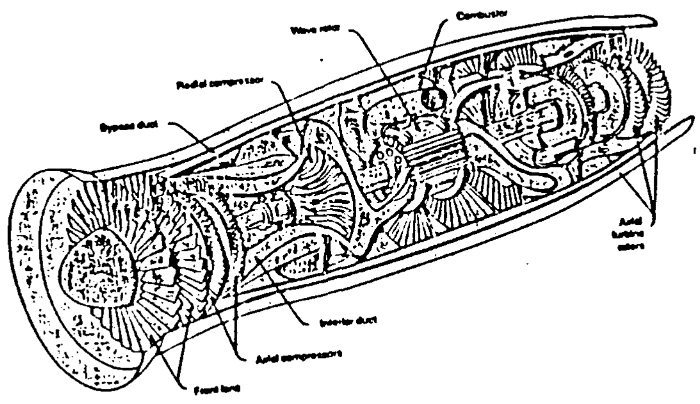


Figure 2. Wave Rotor Turbofan Engine [Ref. 1:p. 293]

Development of wave rotor devices for engines began in the early 1950's. By the early 1960's, gains were achieved in performance through improved seal design and improved timing of the waves arriving at inlet and outlet ports ("tuning") [Ref. 1:p 174]. Few developments were attempted after the 1960's, except by Brown-Boveri in Switzerland. The Brown-Boveri "Comprex" is presently being produced as a commercial supercharger and is available in trucks, heavy equipment and Mazda diesel automobiles. Reference 1 provides a comprehensive review of wave rotor technology.

Wave rotor components have promising applications in small engines in the 600 to 1000 lb. thrust range. Such engines have applications in cruise missiles, remotely piloted vehicles (RPV's), helicopters and small thrust turbofans. The more conventional approach, however, has been to pursue increased turbine inlet temperatures in conventional gas

turbine engines to achieve the desired improved performance. However, in the process, the requirement for turbine cooling also increased. The wave rotor offers the potential to increase performance without the attendant necessity for cooling.

Studies of engine performance indicate that a wave rotor component must be included in the initial cycle optimization rather than be added to an existing engine design. Wave rotor components require, optimally, different compressor and fan pressure ratios. The inclusion of a wave rotor reduces the number of stages required for compression to a constant cycle pressure ratio and allows higher maximum temperatures in the cycle for the same turbine inlet temperature. Performance predictions for gas turbine engines incorporating wave rotor components show decreased specific fuel consumption and increased specific thrust for any fixed value of turbine inlet temperatures.

Figure 3 shows the gains that are thought to be achievable in the performance of small engines [Ref. 1:p. 292]. In Figure 3, "near term improvements" include higher rpm (for a given engine size) and higher turbine temperatures. "New engines" will include composites, ceramics, or new materials. Some current "revolutionary engines" include the compound cycle, the eccentric, the recuperated and the wave rotor engine [Ref. 1:p. 292]. The conventional, eccentric and the recuperated engines are fundamentally limited by the maximum allowable

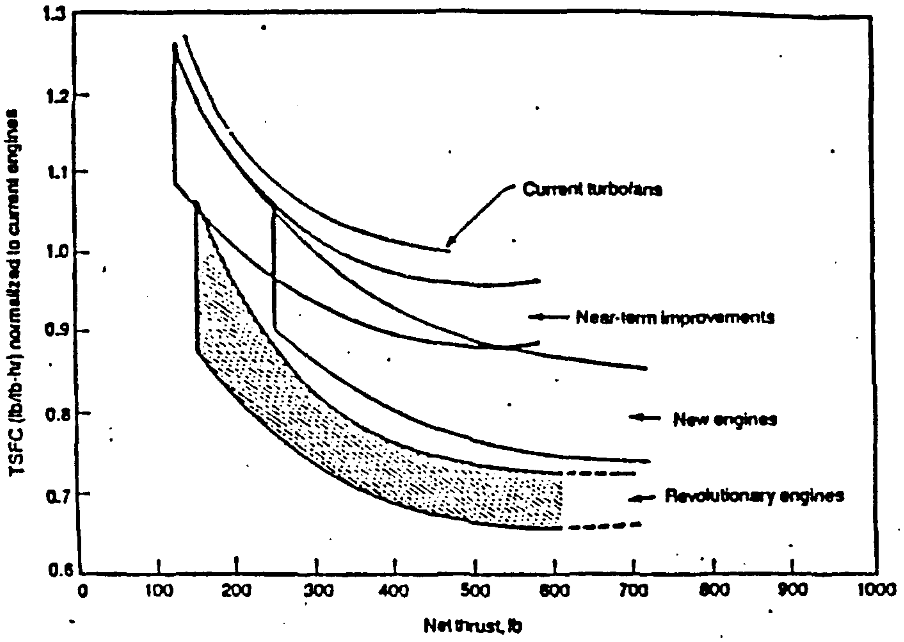


Figure 3. Trends in Engine Development [Ref. 1:p. 292]

turbine inlet temperature (TIT) [Ref. 2:p. 60]. The wave rotor engine is the only engine that promises significantly increased performance without requiring an increase in TIT [Ref. 1:p. 293].

B. ENGINE PERFORMANCE CODES

Two engine performance codes were available for the present study, ENGINE and ONX. There were some differences between the two programs. ENGINE calculated real gas effects based on fuel-air ratio and temperature, while ONX used constant specific heats at the compressor outlet and

downstream of the turbine inlet station. In both programs, the engine performance is obtained by equating power balances across the turbine and compressor sections, and describing an expression for the enthalpy rise across the burner section.

Program ENGINE was developed by A. Mathur [Refs. 3; 4], and uses notation developed by Vavra [Ref. 5], and Vanco [Ref. 6]. The program calculates the performance of a turbofan engine with mixed exhausts with and without the wave rotor component.

Program ONX was developed by J.D. Mattingly [Ref. 7], and it uses methods and notation developed by G. C. Oates [Ref. 8]. The program calculates the performance for a variety of engine configurations including a turbofan with mixed exhausts.

C. WAVE ROTOR INTERNAL FLOW DESCRIPTION

In 1984, A. Mathur developed a FORTRAN program (WRCOMP) to analyze the unsteady processes in wave rotors [Ref. 9]. A detailed description of the analysis and code are provided in Reference 9. WRCOMP uses a 1-D, random choice method (RCM) to solve the hyperbolic, nonlinear conservative system of equations by solving localized Riemann problems. Shocks have sharp resolution. Discontinuities are modelled as jumps. The code is first order accurate in time and it is unconditionally stable. Further explanations and details on the Random Choice

Method are given by Glimm [Ref. 10], Chorin [Ref. 11], and Sod [Ref. 12].

WRCOMP calculates the unsteady process inside the wave rotor, outputs the inlet and outlet port opening and closing times required for correct design, and computes the performance of the wave rotor given the total pressure ratio (TPR), the static pressure ratio (SPR), the exit Mach number (AMEXIT) and the (initial) reference density inside the wave rotor. The performance calculations depend on a favorable comparison for the mass flow rate into and out of the wave rotor component. The outputs from WRCOMP are used as inputs to engine performance calculations performed using the ENGINE code.

III. ENGINE PERFORMANCE CODE

A. DESCRIPTION OF THE CODE

A complete listing, procedural guide and sample results for ENGINE are provided in Appendix A. The code incorporates modular subroutines to calculate each section of the engine cycle. A complete derivation of the code can be found in Reference 4.

The following changes were made to improve the utility of the code:

(1) The original data statement format was modified for a user input interface;

(2) A pop-up menu depicting the configurations was added (see Appendix C for details);

(3) An altitude table was incorporated to automate ambient conditions using Reference 13;

(4) A graphics plotting routine was added using GRAFkit software, Reference 14;

(5) A loop was constructed to increment for:

a) bypass ratio,

b) fan pressure ratio,

c) compressor pressure ratio,

d) LPT exit Mach.

A complete listing of the input parameters and a sample output result appears in Appendix A.

ENGINE (and ONX) computes engine performance in terms of uninstalled specific thrust and specific fuel consumption.

The uninstalled specific thrust is defined as the thrust per unit mass flux through the engine and determines the engine size. The specific fuel consumption is the mass flow rate of fuel per unit thrust and determines the range. A power balance is written for each turbine and compressor spool. An enthalpy rise is written for the flow through the combustor. ENGINE uses loss coefficients and efficiencies to account for component performance. The program also uses real gas effects based on fuel air ratios and temperature. In contrast, ONX uses pressure ratios and polytropic efficiencies to account for component performance and assumes constant specific heats across the compressor (Cp_c) and downstream of the combustor (Cp_t). A necessary step, prior to the use of ENGINE, or comparison between ENGINE and ONX, was to reproduce the results reported by A. Mathur in Reference 4. These results were successfully duplicated using the inputs given in Reference 4 and Appendix A.

To achieve an accurate comparison between ENGINE and ONX, the polytropic efficiencies required in ONX were computed from the outputs of the ENGINE code using the component efficiencies, the actual and isentropic pressures and temperatures, and gamma. A sample calculation is provided in Appendix A.

B. COMPARISON WITH ONX CODE

Since engine cycle calculations are greatly affected by the inputs, especially the component efficiencies, an accurate

comparison between ENGINE and ONX could only be performed if the flight parameters, the design choices, the design limitations and the component efficiencies were the same. Flight parameters include the flight Mach number and the altitude. Design choices include the maximum temperature in the burner, the lower fuel heating value, the maximum turbine inlet temperature, the low pressure turbine exit Mach number, the compressor pressure ratio, the fan pressure ratio, the bypass ratio, etc. Fans, compressors, burners, turbines and nozzles are characterized by component efficiencies, which are the compliments to the loss coefficients. The simplest test case for comparison of ENGINE and ONX was the non-afterburning turbojet. ENGINE calculated the turbojet case by setting the bypass ratio equal to zero.

1. *TURBOJET*

With all input and component performance parameters matched, the results of ENGINE were compared to those of ONX for the case of the turbojet. The results are shown in Table I. For the ONX (1) case, the Cp_c and Cp_t were selected to be in the range that Mattingly referred to as "typical" values [Ref. 7:pp. 116, 438]. The ONX (2) values for Cp_c and Cp_t were the result of numerous iterations using ONX to try to achieve agreement with ENGINE. The ONX (3) values for Cp_c and Cp_t were obtained by computing the average values for Cp_c and Cp_t from

TABLE I

TURBOJET PERFORMANCE COMPARISON

M_0 = 0.79, h=35000 ft.

ENGINE: F/m = 41.88 lbf/(lbm/sec)
SFC = 0.899 (lbm fuel/hr)/lbf thrust

	ONX (1) Typical Specific heats	ONX (2) Values req'd for agreement	ONX (3) Average from ENGINE results
Cp_c	0.235	0.235	0.2434
Cp_t	0.295	0.251	0.2508
F/m	55.18	40.04	35.94
SFC	0.992	0.954	1.001
% Difference F/m	31.76	4.38	14.18
% Difference SFC	10.34	6.10	11.34

the ENGINE program output data. The differences were seen to be surprisingly large

A comparison table was constructed of the pressure and temperature into each component to trace where the calculations began to diverge. The divergence was found to occur in the enthalpy rise calculation across the combustor. For ONX, the enthalpy rise gives the fuel-air ratio (f) as

$$f = \frac{Cp_t \times T_{t4} - Cp_c \times T_{t3}}{[Cp_f \times T_{tf} + h_v \times \eta_b] - Cp_t \times T_{t4}} \quad (1)$$

and the test case in Table I gave f = 0.0152. For ENGINE, the fuel-air ratio equation was:

$$f = \frac{A(T_{t4}) - A(T_{t3}) + f_i \times [B(t_{t4}) - B(t_{t3})]}{[Cp_f \times T_{tf} + h_v \times \eta_b] - B(T_{t4})} \quad (2)$$

where,

$$A(T) = C_1 \times T - \frac{C_2}{2} \times T^2 + \frac{C_3}{3} \times T^3 - \frac{C_4}{4} \times T^4 + \frac{C_5}{5} \times T^5 \quad (3)$$

and

$$B(T) = D_1 \times T + \frac{D_2}{2} \times T^2 - \frac{D_3}{3} \times T^3 + \frac{D_4}{4} \times T^4 - \frac{D_5}{5} \times T^5 \quad (4)$$

With f_i = 0.0 (f prior to the burner), the test case designated ONX (1) in Table I gave f = 0.0103. Thus ONX calculated a 32.24% higher value for the fuel air ratio than did ENGINE. The question then was, can agreement be obtained between ONX and ENGINE using any reasonable values of the Cp's in ONX. The result of an attempt to achieve such agreement is shown as ONX (2) in Table I. Using Cp_c = 0.235 and Cp_t = 0.251, the disagreement in specific thrust was reduced to 4.4% and in SFC to 6.1%. It is noted here that the Cp value for air at the temperature at the exit of the combustor (1860°R) in the

present example is Cp = 0.271, considerably higher than the value required to approach agreement.

One further attempt was made to compare the code predictions for a turbojet. Values of Cp were calculated at the inlet and outlet stations of the compressors and turbines using the enthalpies calculated at the stations by ENGINE in the expression, h = CpT. An average of the in and out values was then used as an input to ONX. The results are shown as ONX (3) in Table I. The strong sensitivity of the performance to the values used for the specific heats is evident in the comparison of the results of ONX (3) with ONX (2).

The selection of Cp values, therefore, had a significant affect on the performance calculations for specific thrust and specific fuel consumption. In order to examine the sensitivity to specific heats, ONX was run with varying Cp_t and varying Cp_c. Figures 4 and 5 show the affect that Cp_t and Cp_c had on the specific thrust and SFC given by the ONX code. The sensitivity to the selection of the input values for ONX is clearly evident in these figures.

A complete deviation of the real gas equations programmed in ENGINE can be found in Reference 3. Further details can be found in References 5 and 6. In short, the stoichiometric combustion equation for a general hydrocarbon fuel $(CH_2)_n$ is expressed as a polynomial expansion. The constants in the expansion C_1 to C_5 and D_1 to D_5 in Equations

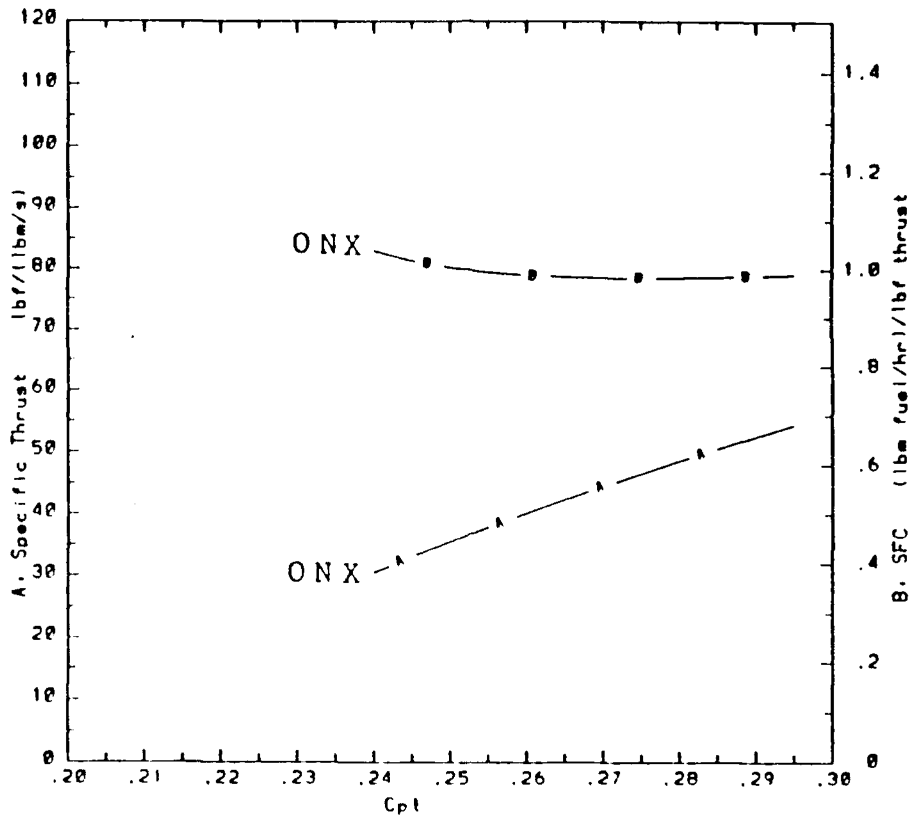


Figure 4. Effect of Cp_t on Specific Thrust and SFC

($\gamma_c = 1.4$, $\gamma_t = 1.350$, $Cp_c = 0.2434$)

(3) and (4) are derived from the constituents of the combustion gases. The final form of the enthalpy rise for the combustor is given as Equation (2). Thus Cp is effectively, and properly, a function of fuel-air ratio and temperature. Since the ENGINE code calculated the enthalpy using the fuel-air ratio and temperature rather than maintaining constant

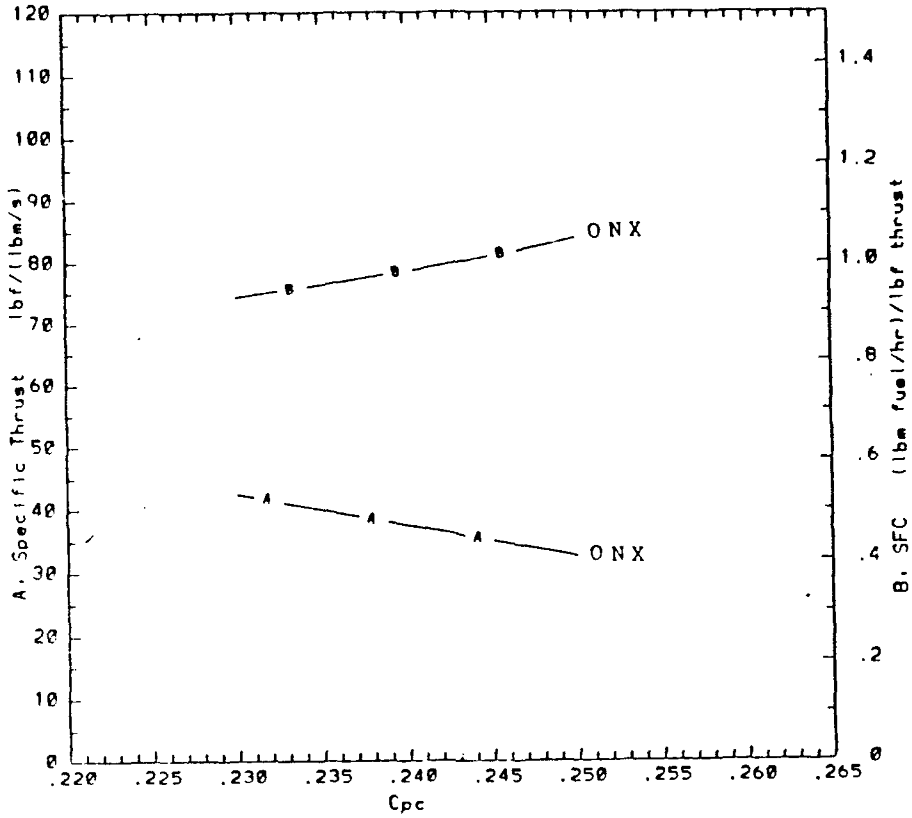


Figure 5. Effect of Cp_c on Specific Thrust and SFC

($\gamma_c = 1.400$, $\gamma_t = 1.350$, $Cp_t = 0.2508$)

values for specific heats, the ENGINE code results were now considered to be more reliable than the ONX results.

2. <u>TURBOFAN</u>

After successfully reproducing A. Mathur's results [Ref. 4:p.7] for a turbofan with mixed exhausts, a comparison between ENGINE and ONX was carried out for the same engine.

The average values of Cp's given by the ENGINE results were input into ONX. The results are presented in Table II. The percent differences in the specific thrust and SFC, predicted by ONX in comparison to ENGINE were 11% and 23% respectively. [Ref. 4:p. 7]

TABLE II

TURBOFAN PERFORMANCE COMPARISON

M_o= 0.79, h=35000 ft.

ENGINE [Ref. 4:P.7]:
F/m = 30.185 lbf/(lbm/sec)
SFC = 0.832 (lbm fuel/hr)/lbf thrust

	ENGINE	ONX Cp_c=0.2434 Cp_t=0.2508
F/m	30.18	33.837
SFC	0.832	1.0926
% Difference F/m	0	10.8
% Difference SFC	0	23.8

The bypass ratio was varied keeping other parameters fixed. Using ENGINE, a very narrow range of bypass ratio was found for which the static pressures of the bypass stream and the core flow could be equal at the constant pressure splitter plate aft of the low pressure turbine. The results in Figure 6 show that common trends were obtained with ENGINE and ONX. As bypass ratio was increased, both SFC and specific thrust decreased. However, the effect on specific thrust was the largest. In fact, only a 5% decrease in SFC could be obtained at the cost of a 20-30% decrease in specific thrust.

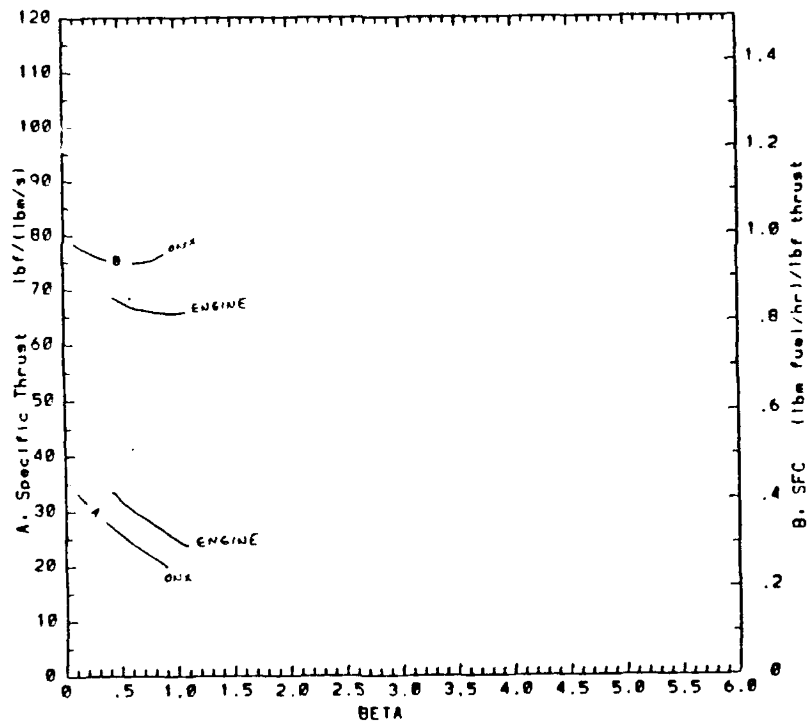


Figure 6. Engine Performance Trends for ONX and ENGINE Varying Bypass Ratio

(M_o = 0.79, h = 35000 ft, π_f = 2.3, π_c = 10.87, TIT = 1860 °R)

To complete the comparison of ENGINE and ONX and to examine the predictions of the ENGINE program, the following parameters were varied and the results obtained for specific thrust and SFC were plotted in the indicated figures: 1) exit Mach number from the LPT (Figure 7); 2) maximum temperature in the combustor (Figure 8); 3) fan pressure ratio (Figure 9); and 4) compressor pressure ratio (Figure 10).

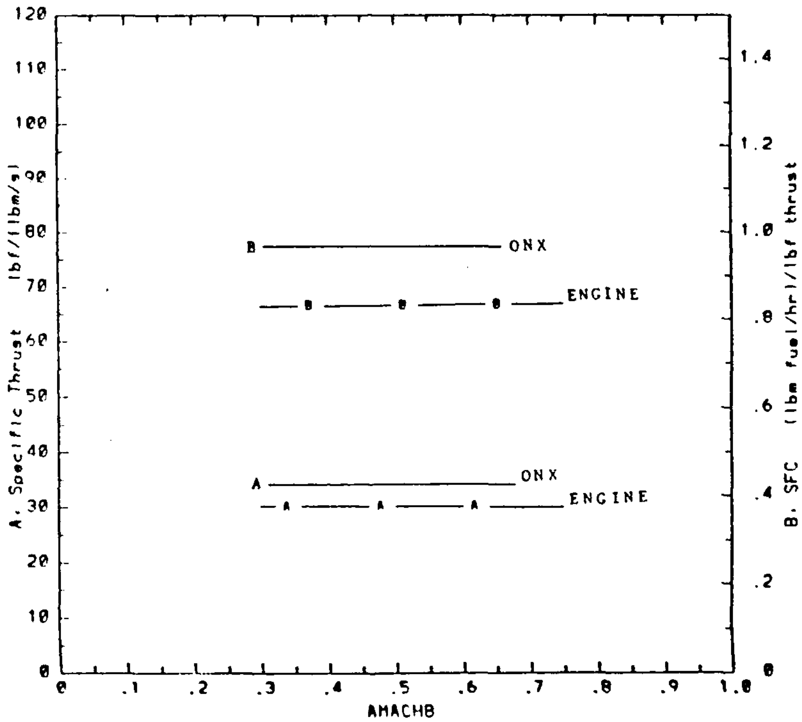


Figure 7. Effect of LPT Exit Mach Number (AMACH8) on Specific Thrust and SFC

(M_o = 0.79, h = 35,000 ft, π_f = 2.3, π_c = 10.87, TIT = 1860°R)

As shown in Figure 7, the LPT exit Mach number had little effect on the specific thrust or on the SFC.

In Figure 8, results are shown for varying turbine inlet temperature obtained using ENGINE with bypass ratio allowed to vary and with ONX with bypass ratio fixed. In the cases shown in Figure 8, the bypass ratio of ENGINE was allowed to vary such that the static pressure at the splitter plate, where the bypass air is mixed with core air, was matched. The mixed exhaust forced higher values of bypass ratio as the TIT was increased. The higher bypass ratio allowed the specific thrust to remain fairly constant. Since significantly more air was bypassed, less fuel was required to produce the same specific thrust as the TIT was increased. Since the bypass ratio increased, the SFC decreased as the TIT increased.

The results from ONX plotted on Figure 8 show the effect of increasing TIT while keeping bypass ratio constant. Fixing the bypass ratio resulted in increased specific thrust but at the price of increased SFC.

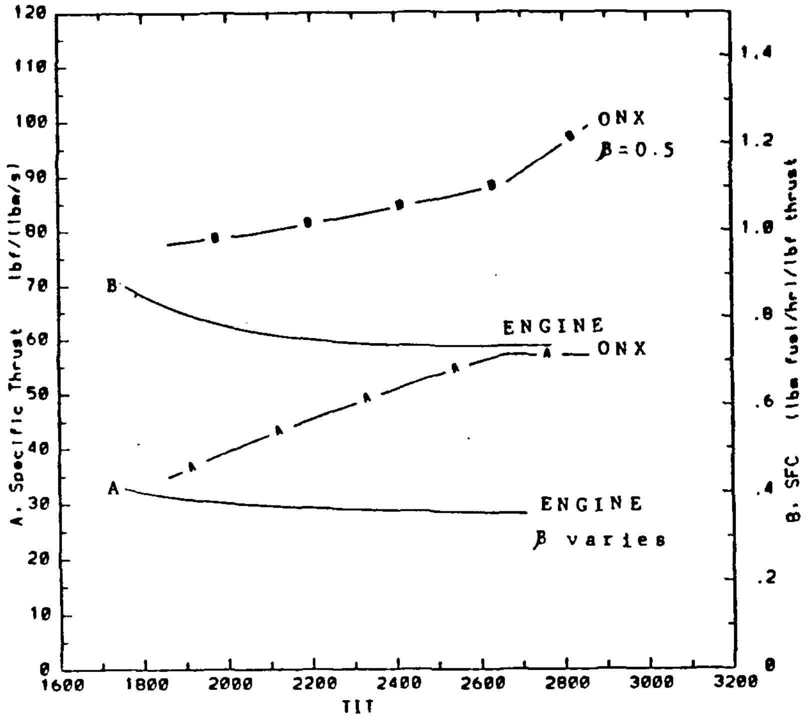


Figure 8. Effect of Varying TIT on Specific Thrust and SFC (M_0 = 0.79, h = 35,000 ft, π_f = 2.3, π_c = 10.87, TIT = 1860°R)

The range of fan pressure ratio was taken to be between 1.0 and 4.0. Figure 9 shows the performance calculated as the fan pressure ratio was varied between 1.0 and 3.4. ENGINE required that the bypass ratio vary to satisfy the equal-static-pressure condition at the splitter plate. A very narrow band of fan pressure ratios was obtained if the bypass ratio was also specified. ONX produced a broader range of values for a specified bypass ratio.

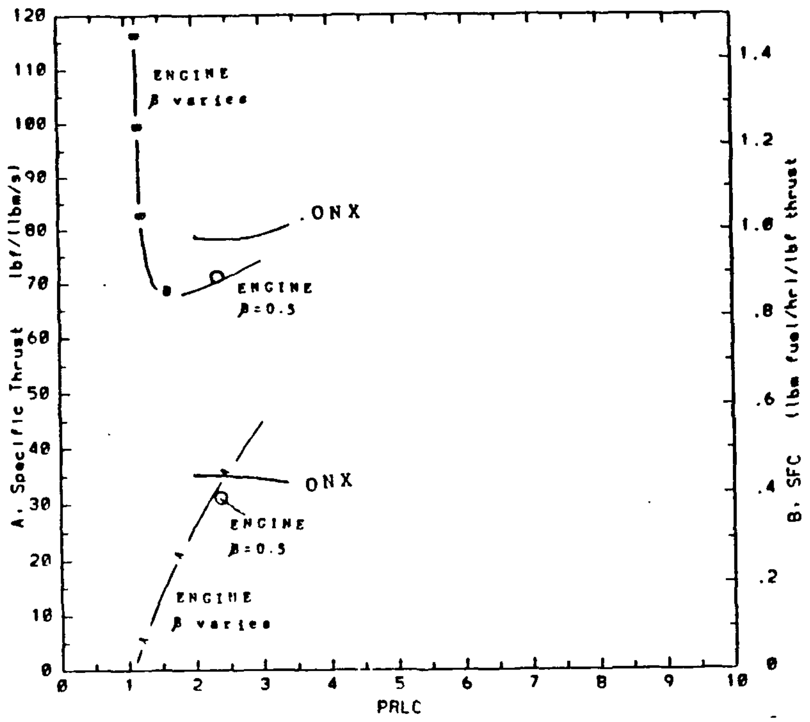


Figure 9. Effect of Varying PRLC on Specific Thrust and SFC (M_o = 0.79, h = 35,000 ft, π_c = 10.87, TIT = 1860°R)

The result of varying compressor pressure ratio is shown in Figure 10. The bypass ratio was fixed at β = 0.5. ENGINE predicted a minimum SFC for compressor pressure ratio range between 11.5 and 12.5 (whereas the minimum predicted by ONX was greater that 20). It is clear from the curves in Figure 10 that a pressure ratio less than that for optimum SFC would be chosen to obtain a significant improvement in specific thrust.

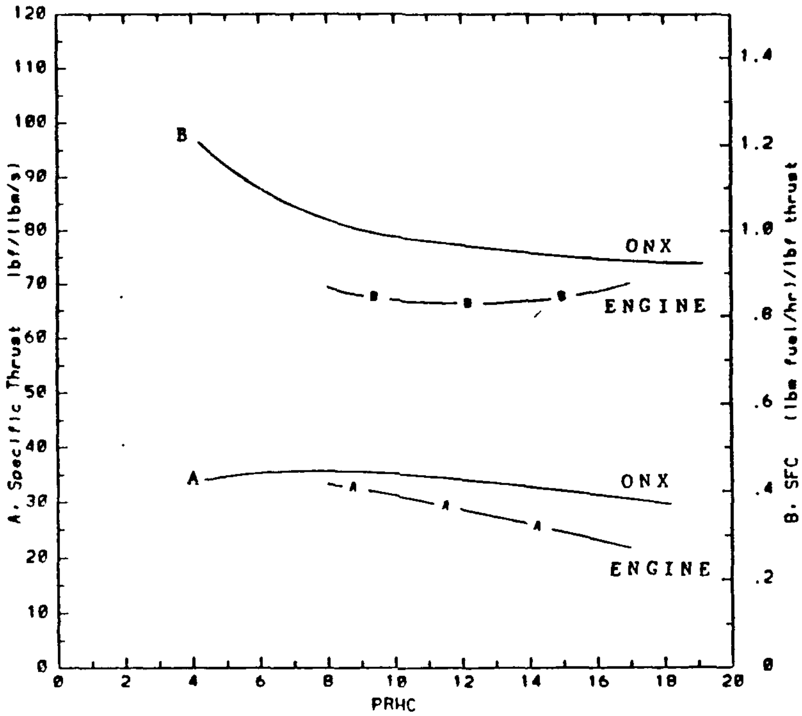


Figure 10. Effect of Varying PRHC on Specific Thrust and SFC (M_o = 0.79, h = 35,000 ft, π_c = 2.3, TIT = 1860°R)

IV. WAVE ROTOR ANALYSIS

A. SUMMARY DESCRIPTION

The wave rotor component and the combustion chamber can be considered to be combined to form a two port component called a gas generator. Other models of the wave rotor component are possible, but the gas generator model is the most straightforward. The equations and code for the wave rotor flow modeling used in the WRCOMP program are described in Reference 9. The equations for the wave rotor component as a gas generator included in the ENGINE code are derived in Reference 4. Figure 11 shows an unwrapped wave rotor and a simplified wave diagram [Ref. 1:p. 38]. Reference 1 describes the gas generator process as a filling, emptying, filling and emptying process.

The process can equally be described as two scavenging processes separated by periods of stationary flow within the rotor. There is a low pressure scavenge at station (4) and a high pressure scavenge at station (2). The combustion exhaust gas initially compresses the air inside the wave rotor and then is scavenged to (4) by the incoming air at the low pressure port (1). When the high pressure exhaust port opens, the high pressure gas inside the wave rotor exits to the HPT and effectively scavenges the wave rotor [Ref. 15:p. 63].

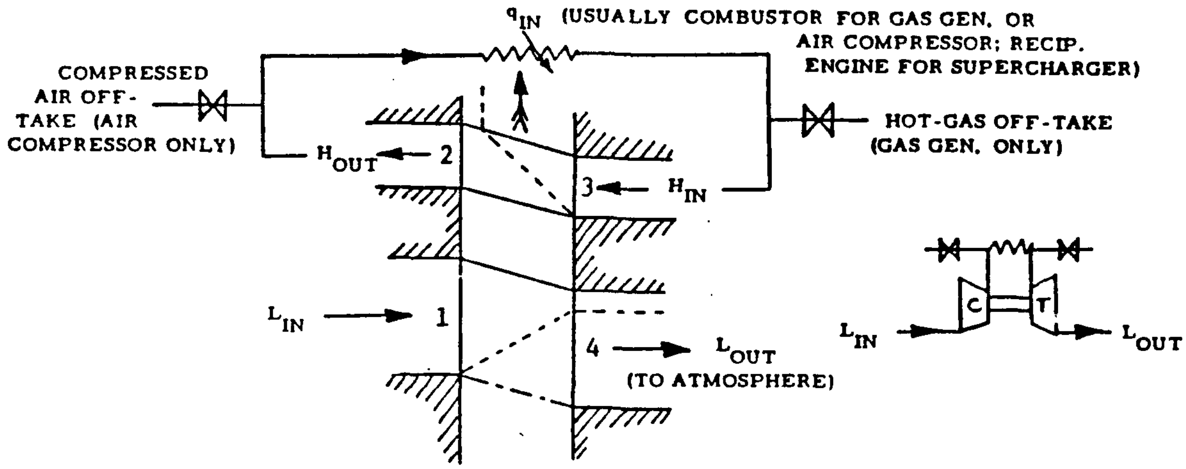


Figure 11. Unwrapped Wave Rotor Component [Ref. 1:p. 38]

In the ENGINE program, the gas generator is modelled as an internal combustion process in which a gain in total pressure and a rise in the total temperature is achieved. The performance of the gas generator is expressed as a function of the static pressure ratio, the total pressure ratio and the total temperature ratio across the inlet ports [Ref. 3:p. 14].

B. INTERFACE WITH ENGINE PERFORMANCE CODE

WRCOMP can be run alone or in conjunction with the ENGINE code. Since the VAX system operates in windows, both programs can be run simultaneously. Both procedures for running WRCOMP are outlined in Appendix B. The original code used data statements in SI units. It was modified to accept input pressures in psi, temperatures in °R, density in lbm per ft^3,

gamma, fuel air ratio (f), and wave rotor exit Mach number. In the coding, the analysis is nondimensionalized to the input parameters. An output file, WRPERF1.OUT, was incorporated to store the input parameters and to convert the English units to SI units. Routines were added to compute the procedures outlined in Reference 4 and to send the output to WRPERF1.OUT. The DISSPLA plotting routines were disabled. In practice, WRCOMP is run until the mass flow into the wave rotor is equal to the mass flow out. Adjustments are made progressively to RREF, the reference density inside the wave rotor, until the mass flows are equal. For 1681 iterations, a CFL number = 0.6, a grid cell width of 0.01, and performing 20 Riemann iterations, the total run time was about 14 minutes on the VAX. An example output is presented in Appendix B.

From experience with WRCOMP and ENGINE, it is easier to keep the overall pressure ratio constant for a series of test cases. By keeping the overall pressure ratio constant, the outlet pressure from the HPC stays the same. If the outlet pressure from the HPC is the same and the performance parameters for the wave rotor are kept the same, then the value for the reference density which satisfies the mass flow solution for WRCOMP also stays the same.

In practice, it is better to vary the overall pressure ratio and maintain the same turbine inlet temperature. This allows optimizing the fan, the compressor and the wave rotor based on realistic cycle constraints.

V. LOW BYPASS TURBOFAN CYCLE CALCULATIONS

A. GENERAL CONSIDERATIONS

The present work has been concentrated on the low bypass turbofan engine, and what performance improvements can be obtained by incorporating a wave rotor component. The initial calculations have been made without reference to a specific vehicle and mission. Considerations which underlie the calculations and which limit their scope are discussed in the following paragraphs.

The characteristics of low bypass turbofans provide a reasonable match to the performance requirements for cruise missiles and RPV's. However, the design limitations for the components of a cruise missile may vary significantly from those of an RPV, even though the two engines may have very similar thrust requirements. A cruise missile may require compact packaging and "low cost design" as it is considered a "throw-away" or one-time-use engine, thereby relaxing the TIT and cooling requirements. An RPV may require a lower TIT to prolong its service life.

Some limitations are inherent in the bypass engine type. A turbofan with mixed exhausts may be bypass ratio limited. The bypass air is mixed with the core air at the splitter plate, where the static pressures must match. If the Mach number of the bypass air is too low, the losses are too great.

Reasons for mixing the exhaust streams are to reduce engine noise and to reduce the IR signature by reducing the temperature of the core air by mixing with the cold bypass air. There is an improvement in performance ideally if mixing occurs internally when there is a large temperature difference between the two streams. [Ref. 8:pp. 165, 172]

The present representation of the wave rotor component is somewhat limited. The modeling associated with instantaneous port opening and closing may require further investigation. Slow port openings may provide more efficient wave rotor operation at length to width ratios below three and stagnation pressure losses are predicted to increase as the length to width ratio increases from three to six [Ref. 1:p. 236].

In performing cycle calculations, care must be taken to use realistic inputs for the wave rotor pressure ratio. Reference 1, p. 18, suggests some practical limits. As the static pressure ratio (SPR) increases, the efficiency of the wave rotor decreases. For good efficiency, total pressure ratios should be about 2.0. In the gas generator model, or pressure gain combustor, the total pressure ratio (TPR) should range from 1.1 to 2.0, corresponding to a total temperature (TTR) range between 1.5 to 3.8 [Refs 14; 15:p. 707].

The treatment of the wave rotor as an ideal component in the engine cycle is a limitation which eventually must be removed. The engine SFC and specific thrust depend greatly on the component efficiencies used in the performance

calculations. Their affects are not always obvious. A key parameter in turbofan engine design is the bypass ratio. As the bypass ratio increases, specific thrust decreases but SFC decreases to a minimum and then increases [Ref. 8:p. 163]. The optimum arrangement becomes dependent on the efficiencies attained by the individual components.

B. RESULTS

A study was conducted using WRCOMP and ENGINE to compare a turbofan with mixed exhausts and incorporating a wave rotor, with the baseline engine operating with the same turbine inlet temperatures. Table III lists the input parameters used in the comparison. A list of the component efficiencies is given in Appendix A.

In maintaining a constant TIT, the effect of adding a wave rotor component to the turbofan was to change both the SFC and the specific thrust. The results obtained for an engine designed with a wave rotor component operating in a gas generator mode are shown in Figure 12.

For each run the fan pressure ratio (π_f) was fixed and the compressor pressure ratio (π_c) was varied. Initially, the bypass ratio was allowed to vary to satisfy the equal-static-pressure requirement at the splitter plate, for a mixed exhaust. The calculated bypass ratio for all the test cases ranged broadly from zero to three. The overall pressure ratio was varied by fixing the fan and the wave rotor pressure

TABLE III

RANGES OF INPUT PARAMETERS IN ENGINE COMPARISON

M_o = 0.79 h = 35000 ft TIT = 1860 M_{LPT} = 0.4	BASELINE TURBOFAN	TURBOFAN WITH WAVE ROTOR
β (bypass ratio)	0.5	varied
π_f	2.3	2.3 - 3.8
π_c	10.87	3.4 - 10.87
π_{OPR}	25.0	15 - 32
π_{wr}	--	1.6
Static Press. Ratio	--	0.4

ratios and varying the compressor pressure ratio. The fan pressure ratio was then increased. At each fan pressure ratio, a specific range of values was obtained for the compressor pressure ratio for which valid solutions existed. As the fan pressure ratio was increased, the feasible range of values for the compressor pressure ratio decreased (the best performance range for the overall pressure ratio was from 22 to 26), and the bypass ratio decreased. For example, for approximately the same TIT = 1860 and π_{OPR} = 23.6 and for π_f = 2.3 then π_c = 6.4 and β = 1.802; for π_f = 2.8 then π_c = 5.3 and β = 1.144. With the wave rotor incorporated, the SFC was consistently lower than the baseline turbofan. The specific

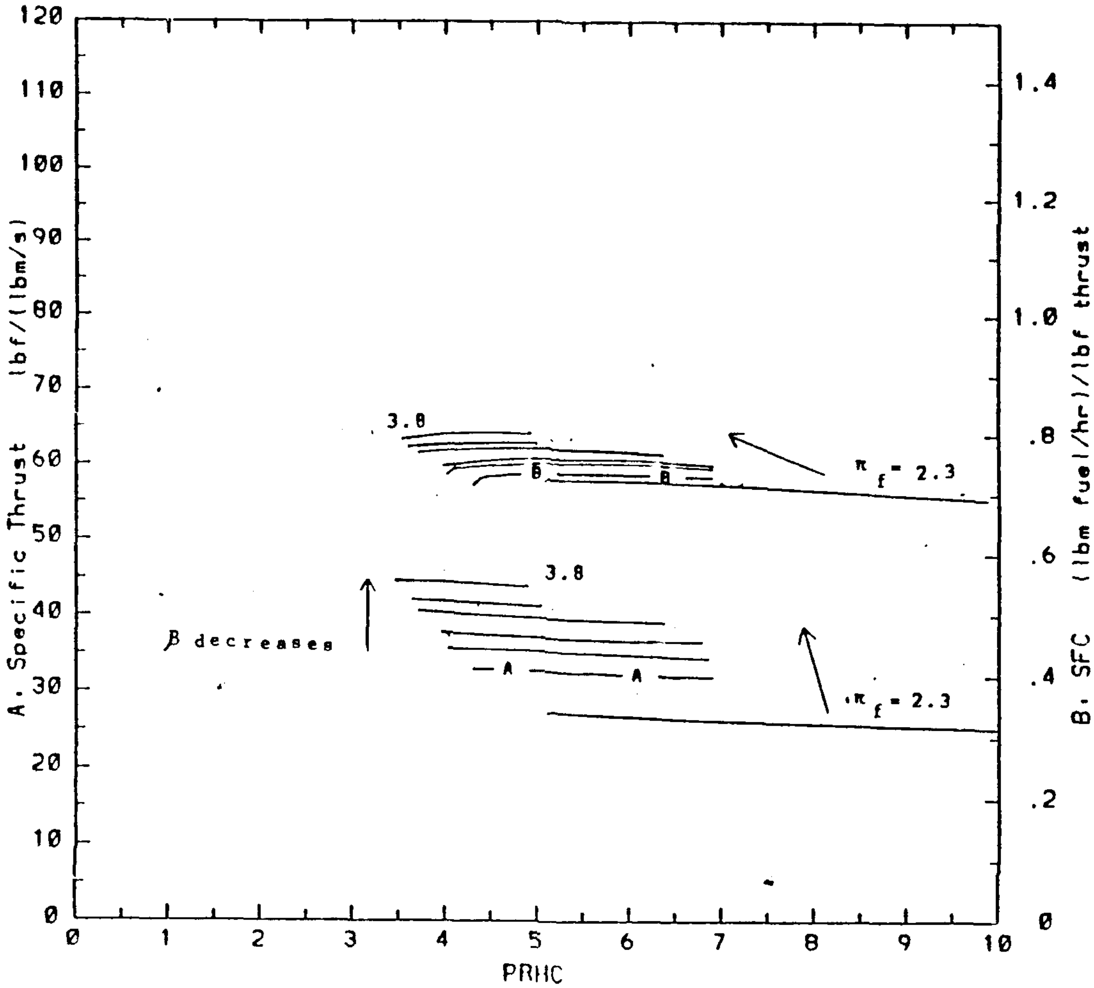


Figure 12. Effect of Varying PRHC and PRLC for a Turbofan with a Wave Rotor

(M_o = 0.79, h = 35,000 ft, AMACH8 = 0.4, β varies)

thrust was also higher than the baseline engine except for the case of π_f = 2.3.

Since a large bypass ratio may not be acceptable for a cruise missile (a larger bypass ratio would require a larger fan size which may not meet the packaging requirements), the

bypass ratio was set to β = 0.5, and the runs repeated. The results are given in Figure 13. A much smaller range of feasible compressor ratios resulted.

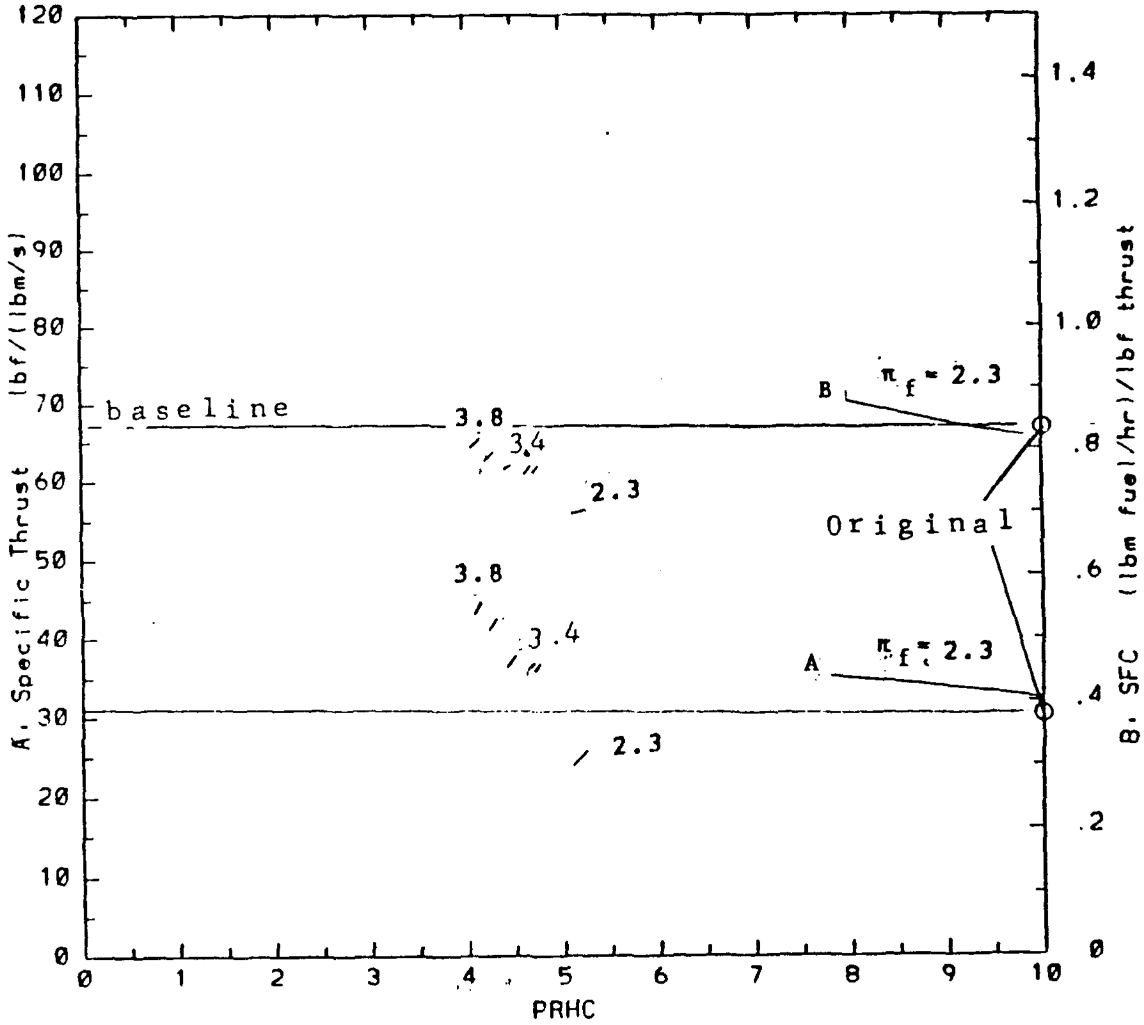


Figure 13. Effect of Varying Compressor Pressure Ratio at β = 0.5 for a Turbofan

(M_o = 0.79, h = 35,000 ft, AMACH8 = 0.4, β varies)

Table IV shows the trade-off between % change in SFC and specific thrust (referred to the baseline turbofan engine) as the fan pressure ratio was changed.

TABLE IV

PERCENT DIFFERENCE FROM BASELINE ENGINE

FAN PRESSURE RATIO	% DIFFERENCE SFC	% DIFFERENCE F/m
2.3	-15.62	-16.83
2.8	-10.22	+ 9.01
3.0	- 8.05	+19.38
3.2	- 7.33	+23.79
3.4	- 5.65	+32.07
3.6	- 4.09	+39.76
3.8	- 3.12	+44.86

Since it was not obvious that the improved performance was due to the wave rotor, the baseline engine performance was calculated as the fan pressure ratio was varied from 2.0 to 3.5. The results are shown in Figure 14. It can be seen in Figure 14 that, as the fan pressure ratio was increased, the SFC increased very slightly and the specific thrust was also almost constant. Thus the improved performance in Table IV was a direct result of the wave rotor.

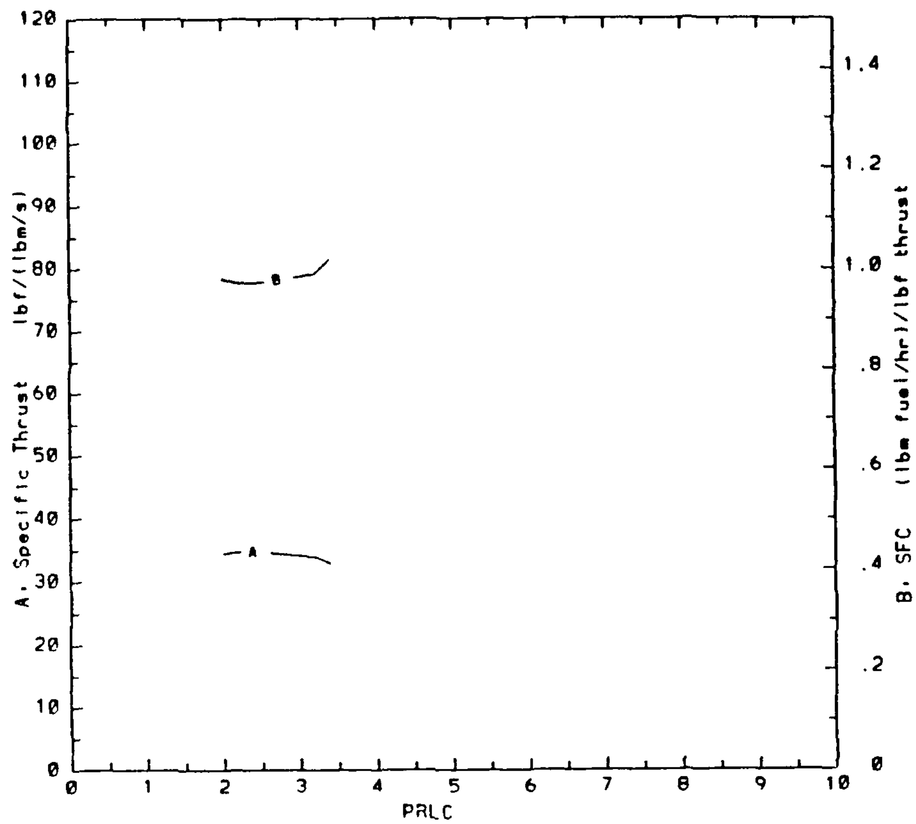


Figure 14. Effect of Varying the Fan Pressure Ratio for the Baseline Turbofan

Table ˝ summarizes the results of the comparison of the baseline turbofan with an engine that incorporates a wave rotor component. The baseline performance is shown compared to the wave rotor turbofan with the same turbine inlet temperature (TIT), the same bypass ratio (β), or the best specific thrust (ST). Of interest to a missile application, a 9% increase in specific thrust and a 7.3% decrease in SFC was obtained by incorporating a wave rotor into the baseline

turbofan and increasing the fan pressure ratio to 2.8, while maintaining $\beta = 0.5$.

Table V also infers other options available in particular applications. For example, if the capture area was not a design limitation, a higher bypass ratio ($\beta = 0.91$) and fan pressure ratio ($\pi_f = 3.0$) could improve the specific thrust (ST) by 16.5% and decrease the SFC by 6.2% while operating at the same TIT = 1860. Similar improvements in specific thrust (ST) and SFC were obtained for lower values of TIT at the specified π_f, as shown in the "best ST" entry in Table V.

TABLE V

COMPARISON OF BASELINE TURBOFAN WITH WAVE ROTOR TURBOFAN

CONFIG	π_f	π_c	π_{opr}	β	TIT ^oR	ST $\frac{lbf}{\frac{lbm}{sec}}$	SFC $\frac{\frac{lbm}{hr}}{lbf}$	% DIFF ST	% DIFF SFC
BASE-LINE	2.3	10.8	25.0	.50	1860	30.2	.832	--	--
WR:									
SAME TIT	2.3	6.4	23.6	1.8	1856	25.6	.727	-15	-12
SAME β	2.3	5.0	18.4	.50	1443	26.1	.744	-13	-11
BEST ST	2.3	4.4	16.2	.03	1267	26.2	.744	-13	-11
SAME TIT	2.8	5.3	23.7	1.1	1876	32.6	.762	8.0	-8.4
SAME β	2.8	4.6	20.6	.53	1624	32.9	.771	9.0	-7.3
BEST ST	2.8	4.0	17.9	.06	1409	33.1	.773	9.6	-7.1
SAME TIT	3.0	4.9	23.5	.91	1859	35.2	.780	16.5	-6.2
SAME β	3.0	4.4	21.1	.48	1665	35.4	.782	17.3	-6.0
BEST ST	3.0	4.0	19.2	.17	1511	35.6	.780	17.8	-6.2
SAME TIT	3.8	3.9	23.7	.44	1874	44.3	.832	46.7	0
SAME β	3.8	4.0	24.3	.53	1923	44.2	.832	46.5	0
BEST ST	3.8	3.8	23.1	.36	1825	44.3	.832	46.9	0

VI. CONCLUSIONS AND RECOMMENDATIONS

In the present study, the computer codes ENGINE and WRCOMP were installed and successfully operated on a VAX computer system. Modifications were made to enhance the utility of the programs and the results reported by the codes' author, A. Mathur, were reproduced.

Comparisons between the predictions of ENGINE and ONX (by J. Mattingly) led to the conclusion that the use of constant specific heats in ONX leads to less accurate results than can be obtained with ENGINE, which includes real gas behavior throughout. Indeed, the predictions of ONX were found to be highly sensitive to the input values of specific heats.

In first attempts to use the codes to examine the potential benefits of incorporating a wave rotor component in a mixed-exhaust by-pass fan engine, it was found that benefits were available which could not be obtained by varying the conventional engine components. For example, while maintaining the same by-pass ratio, a 9% increase in specific thrust and 7.3% decrease in SFC were predicted if, in addition to incorporating the wave rotor, the fan pressure ratio was increased from 2.3 to 2.8. Increases in specific thrust of almost 47% could be obtained with no change in SFC by increasing the fan pressure ratio to 3.8.

These results show that engines incorporating wave rotors can have improved performance over conventional gas turbines which are limited to the same turbine inlet temperature levels. Further extensions of the ENGINE code

(1) to include other engine types

(2) to include friction and heat transfer in the wave rotor simulation

are recommended. Also, studies to examine the effects of varying static pressure ratio and stagnation pressure ratio across the wave rotor, and of increasing turbine inlet temperature, should be carried out.

Finally, an experimental program to validate the wave rotor flow predictions and performance levels used in cycle studies, is required.

APPENDIX A

PROGRAM ENGINE

A. PROCEDURES

1. Log on to the VAX

2. At the DCL prompt type:

```
$gksetup................initialize GRAFkit software
$uis    ................sends plots to screen
$run ENGINE1A...........runs ENGINE program
           ...........select options from screen
```

3. To make changes in ENGINE1A.FOR:

```
$gksetup    .............only required once
$uis        .............
$edit ENGINE1A.FOR ....to enter edit mode
.....
*C        .............to enter full page editing
.....
<CNTRL/Z> .............exits edit mode
*exit     .............saves changes, DCL prompt returns
$FORTRAN ENGINE1A .....compiles FORTRAN code
$link ENGINE1A,'GKL' ..links ENGINE1A and GRAFkit
$run ENGINE1A
          ..............follow screen instructions
```

4. To send your results to a laser printer:

```
$gksetup    .......initialize GRAFkit
            ........environment
$ln03     .............sends output to laser printer
$define gk_out plot.dat .sends output to <fn>.<ft>
          .............where plot.dat is your output
          .............file name
$ln03s_sp ...............output to laser for square
          .............plot
$run ENGINE1A ...........follow screen instructions
          ................
$print/passall/que=ln03 plot.dat
          .............plot.dat is the same file above
```

Note: Since the results are sent to an output file, "plot.dat", they will not appear on the screen.

B. POLYTROPIC EFFICIENCY SAMPLE CALCULATIONS

The following sample calculations use the ENGINE output sample results from Section C in Appendix A.

Compressor efficiency:

In terms of cycle parameters,

$$\eta_c = \frac{ideal\ work\ interaction}{actual\ work\ interaction} \qquad A(1)$$

for a given pressure ratio. In terms of cycle parameters,

$$\eta_c = \frac{\pi_c^{\frac{\gamma_c-1}{\gamma_c}}-1}{\tau_c-1} \qquad A(2)$$

Compressor polytropic efficiency:

For a given differential pressure ratio,

$$e_c = \frac{ideal\ work\ interaction}{actual\ work\ interaction} \qquad A(3)$$

The relationship with compressor efficiency is.,

$$\tau_c = \pi_c^{\frac{\gamma_c-1}{\gamma_c \cdot e_c}} \qquad A(4)$$

so that

$$\ln \tau_c = \frac{\gamma_c-1}{\gamma_c \cdot e_c} \cdot \ln \pi_c \qquad A(5)$$

or

$$e_c = \frac{\gamma_c-1}{\gamma_c} \cdot \frac{\ln \pi_c}{\ln \tau_c} \qquad A(6)$$

For the fan (LPC):

From ENGINE1.DAT:

$$\gamma_c = 1.4004$$
$$\pi_f = \frac{11.7}{5.1} = 2.2941$$
$$\tau_f = \frac{582}{444} = 1.3108$$

giving, with Eq. A(6),

$e_c = 0.8772$

For the high pressure compressor (HPC):

From ENGINE1.DAT:

$$\gamma_c = 1.3870$$
$$\pi_c = \frac{127.0}{11.7} = 10.8547$$

giving, with Eq. A(6),

$e_c = 0.9132$

Turbine efficiency:

For a given pressure ratio,

$$\eta_t = \frac{\textit{actual work interaction}}{\textit{ideal work interaction}} \qquad \text{A(7)}$$

In terms of cycle parameters,

$$\eta_t = \frac{1 - \tau_t}{1 - \pi_t^{\frac{\gamma_t - 1}{e_t \cdot \gamma_t}}} \qquad A(8)$$

Turbine polytropic efficiency:

For a given differential pressure ratio,

$$e_t = \frac{\textit{actual work interaction}}{\textit{ideal work interaction}} \qquad A(9)$$

The relationship to turbine efficiency is

$$\tau_t = \pi_t^{\frac{e_t \cdot (\gamma_t - 1)}{\gamma_t}} \qquad A(10)$$

so that

$$\ln \tau_t = e_t \cdot \frac{\gamma_t - 1}{\gamma_t} \cdot \ln \pi_t \qquad A(11)$$

or

$$e_t = \frac{\gamma_t}{\gamma_t - 1} \cdot \frac{\ln \tau_t}{\ln \pi_t} \qquad A(12)$$

For the high pressure turbine (HPT):

From ENGINE1.DAT:

$$\gamma_t = 1.3436$$
$$\pi_t = \frac{22.8}{123.2} = 0.1851$$
$$\tau_t = \frac{1295}{1860} = 0.6962$$

giving, with Eq. A(12),

$e_t = 0.8392$

For the low pressure turbine (LPT):

$$\gamma_t = 1.3612$$
$$\pi_t = \frac{11.6}{22.8} = 0.5088$$
$$\tau_t = \frac{1104}{1295} = 0.8525$$

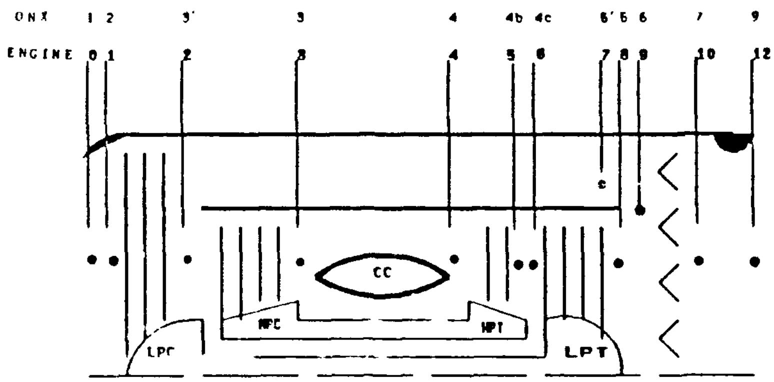
giving, with Eq. A(12),

$e_t = 0.8899$

ENGINE was modified to calculate the polytropic efficiency for comparison with ONX.

C. SAMPLE RESULTS

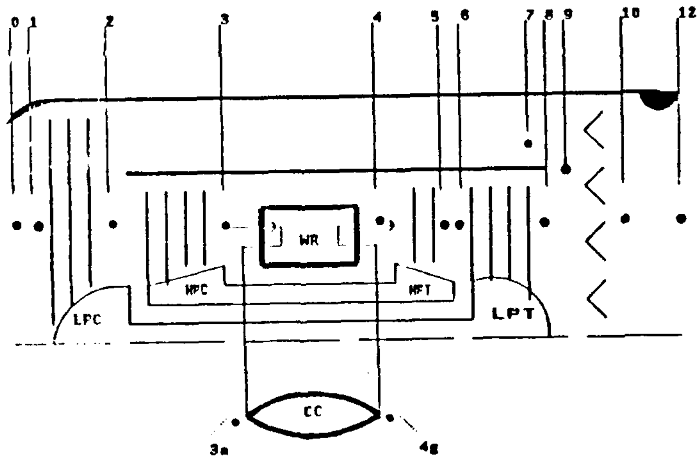
1. NOTATION COMPARISON FOR ONX AND ENGINE



Layout of baseline Turbofan

Station Identification:

0-1	Inlet Section
1-2	Low Pressure Compressor
2-3	High Pressure Compressor
3-4	Main Burner
4-5	High Pressure Turbine
5-6	Constant Pressure Mixer
6-8	Low Pressure Turbine
2-7	Bypass Duct
7,8-9	Constant Area Mixer
9-10	Afterburner
10-12	Nozzle



Wave Rotor as Gas Generator

Station Identification:

0-1	Inlet section
1-2	Low Pressure Compressor
2-3	High Pressure Compressor
3-4	Wave Rotor and Main Burner
4-5	High Pressure Turbine
5-6	Constant Pressure Mixer
6-8	Low Pressure Turbine
2-7	Bypass Duct
7,8-9	Constant Area Mixer
9-10	Afterburner
10-12	Nozzle

2. BASELINE ENGINE CALCULATIONS

a. DATA INPUT

<u>Flight parameters:</u>

$M_o = 0.79$

$h = 35000$ ft.

$p_{so} = 3.468$ psi....\ (calculated within
| the program)
$T_{so} = 394.1\ ^{o}R$ ..../

<u>Design Choices:</u>

$T_{fuel} = 520\ ^{o}R$

FLHV = 18000 BTU/lbm-R

LPT exit Mach number (AMACH8) = 0 4

π_f = PRLC = 2.3

π_c = PRHC = 10.8696

<u>Component Figures of Merit:</u>

η_f = *ETALC* = 0.86
η_c = *ETAHC* = 0.88
η_{HPT} = *ETAHT* = 0.87
η_{LPT} = *ETALT* = 0.90
η_b = *ETACC* = 0.96

<u>Maximum Component Temperatures:</u>

$T_{maxcc} = 1860\ ^{o}R$

$T_{maxAB} = 3400\ ^{o}R$

$T_{maxwr} = 2085\ ^{o}R$

Aircraft System Parameters:

XI = HPT coolant air = 0.0

Bypass ratio = β = 0.5

Component losses:

λ_{BP} = *BPLOSS* = 0.02
λ_{CC} = *CCLOSS* = 0.03
λ_{AB} = *ABLOSS* = 0.02 *if AB off*
= 0.06 *if AB on*

Miscellaneous:

π_{Imax} = PIMAX = 0.97

π_{IDmax} = PIDMAX = 0.97

VCOEFF = 0.97

With the Wave Rotor:

π_{SPR} = SPR = 0.4

π_{TPR} = TPR = 1.6

RREF= reference density in WR = 3.6 kg/m^3

Wr exit Mach number = AMEX = 0.9

π_f = PRLC = 3.8

π_c = PRHC = 4.1

b. ONX OUTPUT

```
                         ON-DESIGN CALCULATIONS

                  TURBOFAN ENGINE WITH MIXED EXHAUST

**********************       INPUT DATA      **********************

    MACH NO    =    .790                ALPHA         =    .00
    ALT (FT)   = 35000.                 PI C'         =   2.30
    TO (R)     = 394.10                 PI D (MAX)    =    .97
    PO (PSIA)  = 3.4680                 PI B          =    .97
    DENSITY    =.00073824               PI N          =    .97
    (SLUG/CUFT)                       EFFICIENCY
    CP C =   .243 BTU/LBM-R             BURNER        =    .96
    CP T =   .265 BTU/LBM-R             MECH HI PR =   1.00
    GAMMA C = 1.400                     MECH LO PR =   1.00
    GAMMA T = 1.350                     LP COMPR (FAN) =    .88 (EC')
    TT4 MAX = 1860.  (R)                HP COMPR       =    .91 (ECH)
    H - FUEL (BTU/LBM) = 18000.         HP TURBINE     =    .84 (ETH)
    CTO              = .0000            LP TURBINE     =    .87 (ETL)
    COOLING AIR #1  =    .00 %          PWR MECH EFF   =  1.00
    COOLING AIR #2  =    .00 %          BLEED AIR      =    .00 %
    PO/P9 =   1.00
    *** MIXER ***                       PI MIXER MAX =   1.00
************************     RESULTS     ************************
    TAU R =   1.125                     AO =   973.1 FT/SEC
    PI R  =   1.509                     VO =   768.7 FT/SEC
    PI D  =    .970                     MASS FLOW = 200.00 LBM/SEC
    TAU L =   5.138                     AREA ZERO  = 10.954 SQFT
    PWR TO=     .00 KW                  AREA ZERO* = 10.507 SQFT
    TAU C'=   1.312                     PT5'/PO =   3.367
    ETA C'=   .8604                     TT5'/TO =   1.476
    PI C    = 25.00                     TAU M1   = 1.0000
    PI C'   = 2.300                     TAU M2   = 1.0000
    TAU C'  = 1.312                     TAU M   = 1.0000
    ETA C'  = .8604                     PI M    = 1.0000
    PI CH   =10.870                     M5      =  .4000
    TAU CH  = 2.108                     M5'     =  .0000
    ETA CH  = .8824                     M6      =  .0000
    PI TH   = .1759                     A5'/A5 =  .0000
    TAU TH  = .6855                     GAMMA M= 1.3500
    ETA TH  = .8669                     CP M    =  .2650
    PI TL   = .6390                     ETA TL =   .8793
    TAU TL  = .9014                     PT9/PO = 3.8694
  WITHOUT AB
    F        =   .0116
    FO       =   .0116
    F/M      = 42.415 LBF/LBM/S
    S        =   .9834 1/HR
    T9/TO    = 2.0535
    V9/VO    = 2.7434
    M9/MO    = 1.9615
    A9/AO    =   .7480
    A9/A8    = 1.2571
    THRUST =   8483. LBF
```

c. ENGINE OUTPUT

```
FPR=2.3  OPR=25.0  TIT=1860.  ABTEMP=3400.  MACH NO=0.79  ALT=35000.

   STATION IDENT: ENGINE INLET, STATIONS 0-1

   TI= 394.  TO= 444.  PI=  3.5  PO=  5.1  FI=0.000  FO=.000  VO= 769.

   STATION IDENT: LP COMPRESSOR, STATIONS 1-2

   TI= 444.  TO= 582.  PI=  5.1  PO= 11.7  FI=.000  FO=.000
   ETALC=0.8600  GBAR=1.4004
   ELPC=0.8757

   STAION IDENT: HP COMPRESSOR, STATIONS 2-3

   TI= 582.  TO=1206.  PI= 11.7  PO=127.0  FI=.000  FO=.000
   ETAHC=0.8800  GBAR=1.3870
   EHPC=0.9144

   STATION IDENT: MAIN BURNER, STATIONS 3-4

   TI=1206.  TO=1860.  PI=127.0  PO=123.2  FI=.000  FO=.010

   STATION IDENT: HP TURBINE, STATIONS 4-5

   TI=1860.  TO=1295.  PI=123.2  PO= 22.8  FI=.010  FO=.010
   ETA=0.8700  GBAR=1.3436
   ET=0.8379

   STATION IDENT: PRE LPT MIXING, STATIONS 3,5-6

   T1=1206.  T2=1295.  TO=1295.  PO= 22.8  F1=.000  F2=.010  FO=.010

   STATION IDENT: BYPASS DUCT, STATIONS 2-7

   TI= 582.  TO= 582.  PI= 11.7  PO= 11.5  FI=.000  FO=.000

   STATION IDENT: LP TURBINE, STATIONS 6-8

   TI=1295.  TO=1104.  PI= 22.8  PO= 11.6  FI=.010  FO=.010
   ETA=0.9000  GBAR=1.3612
   ET=0.8904

   STATION IDENT: CONST. AREA MIXER, STATIONS 7,8-9

   T1=1104.  T2= 582.  TO= 939.  P1= 11.6  P2= 11.5  PO= 11.2

   F1=.010  F2=.000  FO=.007  M1=0.40  M2=0.37  MO=0.40  ARAT=0.38

   STATION IDENT: AFTERBURNER, STATIONS 9-10

   TI= 939.  TO= 939.  PI= 11.2  PO= 11.0  FI=.007  FO=.007

   STATION IDENT: EXHAUST NOZZLE, STATIONS 11-12

   TI= 939.  TO= 697.  PI= 11.0  PO=  3.5  FI=.007  VO=1728.  MO=1.34
   GLOCO=1.3948  PRN=.31661  AEXIT=1294.1

   PERFORMANCE PARAMETERS

   AMACH8= 0.400

   SP.THRUST = 30.18  SFC = 0.832   BETA=  0.500  EXMACH= 1.336
   (LBF/LBM/S)        (LBM/HR/LBF)
```

3. TURBOFAN WITH WAVE ROTOR: ENGINE OUTPUT

```
FPR=3.8  OPR=24.9  TIT=1860.  ABTEMP=3400.  MACH NO=0.79  ALT=35000.
  STATION IDENT: ENGINE INLET, STATIONS 0-1
  TI= 394.  TO= 444.  PI=  3.5  PO=  5.1  FI=0.000  FO=.000  VO= 769.
  STATION IDENT: LP COMPRESSOR, STATIONS 1-2
  TI= 444.  TO= 682.  PI=  5.1  PO= 19.3  FI=.000  FO=.000
  ETALC=0.8600  GBAR=1.3992
  ELPC=0.8841

  STAION IDENT: HP COMPRESSOR, STATIONS 2-3
  TI= 682.  TO=1056.  PI= 19.3  PO= 79.2  FI=.000  FO=.000
  ETAHC=0.8800  GBAR=1.3870
  EHPC=0.9022

  STATION IDENT: WAVE ROTOR, STATIONS 3-4
  TCCIN= 1199.0  TO= 1972.7  TREF= 2197.7
  PIN=  79.2  PO= 122.9  PREF= 183.1
  FIN=0.0000  FO=0.0165  GLOCO=1.371  IDB1=1

  STATION IDENT: HP TURBINE, STATIONS 4-5
  TI=1973.  TO=1649.  PI=122.9  PO= 52.4  FI=.016  FO=.016
  ETA=0.8700  GBAR=1.3263
  ET=0.8551

  STATION IDENT: PRE LPT MIXING, STATIONS 3,5-6
  T1=1056.  T2=1649.  TO=1649.  PO= 52.4  F1=.000  F2=.016  FO=.016
  STATION IDENT: BYPASS DUCT, STATIONS 2-7
  TI= 682.  TO= 682.  PI= 19.3  PO= 18.9  FI=.000  FO=.000
  STATION IDENT: LP TURBINE, STATIONS 6-8
  TI=1649.  TO=1335.  PI= 52.4  PO= 20.6  FI=.016  FO=.016
  ETA=0.9000  GBAR=1.3416
  ET=0.8866

  STATION IDENT: CONST. AR   MIXER, STATIONS 7,8-9
  T1=1335.  T2= 682.  TO=1133.  P1= 20.6  P2= 18.9  PO= 19.4
  F1=.016  F2=.000  FO=.011  M1=0.40  M2=0.19  MO=0.31  ARAT=0.74
  STATION IDENT: AFTERBURNER, STATIONS 9-10
  TI=1133.  TO=1133.  PI= 19.4  PO= 19.0  FI=.011  FO=.011
  STATION IDENT: EXHAUST NOZZLE, STATIONS 11-12
  TI=1133.  TO= 737.  PI= 19.0  PO=  3.5  FI=.011  VO=2232.  MO=1.68
  GLOCO=1.3890  PRN=.18289  AEXIT=1329.8
  PERFORMANCE PARAMETERS


  AMACH8= 0.400
  SP.THRUST = 46.22  SFC = 0.856    BETA=  0.500  EXMACH= 1.678
  (LBF/LBM/S)        (LBM/HR/LBF)
```

D. ENGINE PROGRAM LISTING

```
C       PROGRAM ENGINE
C        DIMENSION XARAY(125),YARAY(125),ZARAY(125),BPR(125)
        REAL XP1(50),YP1(50),YP2(50)
        REAL ST,SPTHR,SFC,PRAT
        INTEGER CONFIG,MCONFIG,II,JJ,I1,J1
        CHARACTER*6 X1LABEL
        CHARACTER*15 Y1LABEL
        CHARACTER*3 Y2LABEL
        COMMON/ETAS/ETALC,ETAHC,ETACC,ETAHT,ETALT,ETAAB,VCOEFF
        COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
        COMMON/BURN/HFUEL,FLHV,TMAXCC,TMAXAB,TMAXWR
        COMMON/WORK1/WORKLC,WORKHC
        COMMON/CONST/CJ,G
        COMMON CONFIG
C****     DEFINITION OF PROGRAM SWITCHES
C**** CONFIG=0:CONVENTIONAL ENGINE; CONFIG=1:WAVE-ROTOR ENGINE.
C**** KODE=0:NON-AFTERBURNING CASE; KODE=1:AFTERBURNER LIT CASE.
C**** KASET=0:BYPASS RATIO (BETA) PRESCRIBED. KASET=1:BETA CALCULATED.
        DATA IDC,IDB,IDT,IDM/0,0,0,0/
        DATA XP1/50*0.0/
        DATA YP1/50*0.0/
        DATA YP2/50*0.0/
        IFLAG=0
C       INPUT DATA (GENERAL)
C *****************************************************************
C       INITIALIZE THE DATA BASE:*****************************************
          WRITE(6,*)'WOULD YOU LIKE TO SEE THE POSSIBLE CONFIGURATIONS
     > FOR ENGINE?? (1=YES 0=NO)'
          WRITE(6,*)'Hit <Return> for next picture and to continue.'
          READ(5,'(I)')IDESIGN
          IF (IDESIGN.EQ.0) GOTO 11
            CALL LIB$SPAWN ('RENDER ENG1.UIS')
            CALL LIB$SPAWN ('RENDER ENG2.UIS')
            CONTINUE
  11      WRITE(6,*)'INPUT DESIRED CONFIGURATION:'
            WRITE(6,*)'CONFIG=(0=CONVENTIONAL 1=WAVE ROTOR)'
            READ(5,'(I)')CONFIG
                MCONFIG=CONFIG
          WRITE(6,*)'KODE=(0=NON A/B 1=A/B ON)'
            READ(5,'(I)')KODE
          WRITE(6,*)'KASET=(0=BYPASS RATIO PRESCRIBED 1=BETA CALCULATED)'
            READ(5,'(I)')KASET
          WRITE(6,*)'WOULD YOU LIKE TO USE YOUR OWN PARAMETERS OR THE
     > DEFAULT'
          WRITE(6,*)' PARAMETERS? (1=OWN,0=DEFAULT)'
            READ(5,'(I)')IQ1
        IF (IQ1.EQ.1) THEN
            CALL FLIGHT(FMACH,ALT,PSO,TSO)
C           WRITE(6,*)'CONFIG=',CONFIG
             IF (CONFIG.EQ.1) THEN
               CALL WRSPECS(SPR,TPR,RREF,AMEX)
             ENDIF
C           WRITE(6,*)'CONFIG=',CONFIG
            CALL DESLIM(TFUEL,FLHV,AMACH8,HFUEL,PRLC,PRHC,PRAT,TPR,
     >CONFIG)
            CALL COMPFOM(ETALC,ETAHC,ETAHT,ETALT,ETACC,ETAAB)
            CALL TMAX(TMAXCC,TMAXAB,TMAXWR)
            CALL ACSYS(CTO,TMAXCC,XI,BETA)
            CALL COMLOSS(BPLOSS,CCLOSS,ABLOSS)
            CALL COMPETA(PIMAX,PIDMAX,VCOEFF)
```

```
        ELSEIF (IQ1.EQ.0) THEN
          CALL BASE1(FMACH,ALT,PSO,TSO,TFUEL,FLHV,AMACH8,HFUEL,
     >PRLC,PRHC,PRAT,ETALC,ETAHC,ETAHT,ETALT,ETACC,ETAAB,EATDB,
     >TMAXCC,TMAXAB,TMAXWR,CTO,XI,BETA,BPLOSS,CCLOSS,ABLOSS,
     >PIMAX,PIDMAX,VCOEFF,SPR,TPR,RREF,AMEX,CONFIG,KODE)
      ENDIF
C-----SET INTIAL CONSTANTS-----------------------------------------
      NN=0
      CJ=778.16
      G=32.174
      FO=0.0
C=================================================================
C-----END OF INPUT DATA
C=================================================================
      OPEN (UNIT=9,FILE='ENGINE1.DAT',STATUS='NEW')
C=================================================================
C *** CALLING SEQUENCE OF COMPONENT SUBROUTINES FOR CHOSEN ENGINE ***
C-----CONFIGURATION
C=================================================================
      WRITE(6,*)'WOULD YOU LIKE TO ITERATE A VARIABLE (1=YES 0=NO)'
        READ(5,'(I)')IANS1
      IF (IANS1.EQ.0) THEN
          GO TO 6
        ELSE
         WRITE(6,*)'INPUT THE DESIRED ITERATION PARAMETER (ie 1)'
         WRITE(6,*)'     1. BYPASS RATIO'
         WRITE(6,*)'     2. FAN PRESSURE RATIO'
         WRITE(6,*)'     3. COMPRESSOR RATIO'
         WRITE(6,*)'     4. LPT EXIT MACH NUMBER'
          READ(5,*)IANS4
           IANSW4=IANS4
         WRITE(6,*)'INPUT MIN VALUE ='
          READ(5,*)SMIN
         WRITE(6,*)'INPUT MAX VALUE ='
          READ(5,*)SMAX
         WRITE(6,*)'INPUT STEP SIZE ='
          READ(5,*)STEP
         OPEN(UNIT=14,FILE='PLOT1.DAT',STATUS='NEW')
         IF (IANS4.EQ.1) THEN
           X1LABEL=' BETA '
C          WRITE(14,'(2X,A,2X,A,2X,A)')'  BETA',' ST  ','  SFC '
          ELSEIF (IANS4.EQ.2) THEN
           X1LABEL=' PRLC '
C          WRITE(14,'(2X,A,2X,A,2X,A)')'  PRLC',' ST  ','  SFC '
          ELSEIF (IANS4.EQ.3) THEN
           X1LABEL=' PRHC '
C          WRITE(14,'(2X,A,2X,A,2X,A)')'  PRHC',' ST  ','  SFC '
          ELSEIF (IANS4.EQ.4) THEN
           X1LABEL='AMACH8'
C          WRITE(14,'(2X,A,2X,A,2X,A)')'AMACH8',' ST  ','  SFC '
         ENDIF
      WRITE(14,'(2X,A,2X,A,2X,A,2X,A,2X,A,2X,A6,2X,A,2X,A)')' BETA ',
     >' PRLC ',' PRHC ',' PRAT ',' TIT ',X1LABEL,' ST  ','  SFC '
              IC2=1
C------START LARGE LOOP ITERATION FOR ITERATION VARIABLE---------
         DO 41 SVALUE=SMIN,SMAX,STEP
           IF (IANS4.EQ.1) THEN
             BETA=SVALUE
            ELSEIF (IANS4.EQ.2) THEN
             PRLC=SVALUE
```

```
          ELSEIF (IANS4.EQ.3) THEN
           PRHC=SVALUE
          ELSEIF (IANS4.EQ.4) THEN
           AMACH8=SVALUE
         ENDIF
         IF (CONFIG.EQ.1) THEN
           PRAT=PRLC*PRHC*TPR
          ELSE
           PRAT=PRLC*PRHC
         ENDIF
C-----WRITE STATION RESULTS TO DEVICE 9 ON FILE ENGINE1.DAT------------
C-----ITERATION PLOT RESULTS ARE SENT TO 14 AND PLOT1.DAT--------------
   6   WRITE(9,1)PRLC,PRAT,TMAXCC,TMAXAB,FMACH,ALT
   1   FORMAT(2X,'FPR=',F3.1,2X,'OPR=',F4.1,2X,'TIT=',F5.0,2X,'ABTEMP=',
     >F5.0,2X,'MACH NO=',F4.2,2X,'ALT=',F6.0/)
       CALL INLET(PSO,TSO,FO,P1,T1,F1,VO,FMACH,PIDMAX)
       CALL COMPR(PRLC,P1,T1,P2,T2,T2IS,F1,F2,IDC)
       CALL COMPR(PRHC,P2,T2,P3,T3,T3IS,F2,F3,IDC)
         IF(CONFIG.EQ.1) GOTO 10
       CALL BURNER(P3,T3,P4,T4,F3,F4,IDB,KODE,CONFIG)
         GOTO 20
c   10 CALL ROTOR(P3,P4,T3,T4,PREF,RREF,TREF,F3,F4,SPR,TPR,AMEX,TCCIN,
c     >IDB,CONFIG)
   10 CALL ROTOR(PRAT,P1,P3,P4,T3,T4,PREF,RREF,TREF,F3,F4,SPR,TPR,AMEX,
     >TCCIN,IDB,CONFIG)
   20 CALL TURBIN(P4,T4,P5,T5,T5IS,F4,F5,H5,PRAT,PRLC,PRHT,PRLT,XI,BETA,
     >KASET,IDT)
       CALL MIXCP(F3,F5,P3,T3,P5,T5,F6,P6,T6,XI,IDM)
       CALL DUCT(P2,T2,F2,P7,T7,F7,H7)
      CALL TURBIN(P6,T6,P8,T8,T8IS,F6,F8,H8,PRAT,PRLC,PRHT,PRLT,XI,BETA,
     >KASET,IDT)
         IF(KASET.EQ.1) GOTO 30
       CALL MIXCA(P7,P8,T7,T8,H7,H8,F7,F8,BETA,AMACH8,PIMAX,T9,F9,P9,
     >AMACH9)
         GOTO 40
   30  CALL MIXCP(F7,F8,P7,T7,P8,T8,F9,P9,T9,BETA,IDM)
   40  CALL BURNER(P9,T9,P10,T10,F9,F10,IDB,KODE,CONFIG)
      CALL NOZZLE(P10,T10,P12,T12,T12IS,F10,PRLC,VEXIT,EXMACH,PSO,IFLAG)
         IF(IFLAG.EQ.1) GOTO 50
       CALL PERF(AMACH8,F10,VO,VEXIT,BETA,SPTHR,SFC,EXMACH)
       WRITE(14,3014) BETA,PRLC,PRHC,PRAT,T4,SVALUE,SPTHR,SFC
C      WRITE(14,'(2X,F6.3,2X,F6.3,2X,F6.3)')SVALUE,SPTHR,SFC
C        WRITE(6,*)'IC2=',IC2
C        WRITE(6,*)'ST=',SPTHR,'SFC=',SFC
               XP1(IC2) = SVALUE
               YP1(IC2) = SPTHR
               YP2(IC2) = SFC
C        WRITE(6,*)'XP1=',XP1(IC2),'YP1=',YP1(IC2),'YP2=',YP2(IC2)
               IF (SVALUE.LE.SMAX) IC2=IC2+1
C-----------RESET THE COUNTER SWITCHES EACH LOOP
               IDC=0
               IDB=0
               IDT=0
               IDM=0
               CONFIG=MCONFIG
C-------CHECK PROPER COUNTER OPERATION--------------------------------
C      WRITE(6,*)'IDC=',IDC,'IDB=',IDB,'IDT=',IDT,'IDM=',IDM
C      WRITE(6,*)'CONFIG=',CONFIG
   41  CONTINUE
C------END LARGE LOOP VARIATION FOR ITERATION VARIABLE----------------
```

```
      END IF
      ENDFILE 14
      REWIND 14
      CLOSE (UNIT=14)
C---------------------------------------------------------------------
C     CALL GRAFKIT PLOT ROUTINE
C---------------------------------------------------------------------
     WRITE(6,*)'WOULD YOU LIKE TO PLOT THE RESULTS? (1=YES 0=NO)'
      READ(5,'(I)')IC5
      IF (IC5.EQ.1) THEN
C----- NOW CALL GRAPHICS ROUTINE
        CALL GRSPLOT1(XP1,YP1,YP2,IANSW4,IC2-1)
      ENDIF
C ***  ***********************************************************
C ***  ***END OF USER-DEFINED CALLING SEQUENCE**********************
C ***  ***********************************************************
      OPEN(UNIT=10,FILE='ENGINE2.DAT',STATUS='NEW')
      WRITE(10,3000)P6,P7,P8,P11
      CLOSE(UNIT=10)
 3000 FORMAT(5X,'P6=',F5.1,5X,'P7=',F5.1,5X,'P8=',F5.1,5X,'P11=',
    >F5.1//)
 3014 FORMAT(2X,F6.3,2X,F6.3,2X,F6.3,2X,F6.3,2X,F6.1,2X,F6.3,2X,
    >F6.3,2X,F6.3)
      GOTO 70
   50 WRITE(6,60)
   60 FORMAT(5X,'NOZZLE PRESSURE RATIO IS NEGATIVE'/)
   70 CLOSE(UNIT=9)
      STOP
     END
C---------------------------------------------------------------------
     SUBROUTINE BASE1(FMACH,ALT,PSO,TSO,TFUEL,FLHV,AMACH8,HFUEL,
    >PRLC,PRHC,PRAT,ETALC,ETAHC,ETAHT,ETALT,ETACC,ETAAB,EATDB,
    >TMAXCC,T[illegible],[illegible]MAXWP,[illegible]CTO,XI,BETA,[illegible]PLOSS,CCLOSS,ABLOSS,
    >PIMAX,PIDMAX,VCOEFF,SPR,TPR,RREF,AMEX,ICONFIG,IKODE)
     COMMON CONFIG
C-----FLIGHT PARAMETERS
      FMACH=0.79
      ALT=35000.
      PSO=3.468
      TSO=394.1
C-----WAVE ROTOR SPECS
      SPR=0.40
      TPR=1.60
      RREF=3.6
      AMEX=0.9
C--------TCCIN IS SOLVED FOR IN ROTOR ROUTINE----------------
C        TCCIN=1213.3
C-----DESIGN/FUEL CHOICES
      TFUEL=520.
      FLHV=18000.
      AMACH8=0.4
      HFUEL=TFUEL*0.5
      IF (ICONFIG.EQ.1) THEN
        PRLC=3.8
        PRHC=4.1
       ELSE
        PRLC=2.3
        PRHC=10.8696
      ENDIF
        IF (ICONFIG.EQ.1) THEN
```

```
          PRAT=PRLC*PRHC*TPR
C         WRITE(6,*)'PRAT=',PRAT,'TPR=',TPR,'CONFIG=',ICONFIG
        ELSE
          PRAT=PRLC*PRHC
        ENDIF
C-----COMPONENT FOM
      ETALC=0.86
      ETAHC=0.88
      ETAHT=0.87
      ETALT=0.90
      ETACC=0.96
      IF (IRODE.EQ.0) THEN
         ETAAB=1.00
        ELSE
         ETAAB=0.96
      ENDIF
C-----MAX COMPONENT TEMPERATURES (R)
C     NOTE: THAXCC AND THAXWR GET RESET TO VALUES CORRESPONDING TO
C           THE WAVE ROTOR ANALYSIS : TMAXCC=TREF, TMAXWR=TREF
      TMAXCC=1860.0
      TMAXAB=3400.
      TMAXWR=2085
C-----AIRCRAFT SYSTEM PARAMETERS
      CTO=0.0
      WRITE(6,*)'INPUT % COOLING OR BASE IT ON TMAXCC:'
      WRITE(6,*)'  (1=INPUT FOR XI    0=BASE XI ON TMAXCC)'
      READ(5,*)ICOOL1
      IF (ICOOL1.EQ.0) THEN
        IF (TMAXCC.GT.2400) THEN
          XI=2.0*(TMAXCC-2400.0)/16000.0
        ELSE
          XI=0.0
        ENDIF
      ELSE
       WRITE(6,*)'INPUT XI'
       READ(5,*)XI
      ENDIF
      BETA=0.5
C-----COMPONENT LOSSES
      BPLOSS=0.02
      CCLOSS=0.03
      IF (IRODE.EQ.0) THEN
         ABLOSS=0.02
        ELSE
         ABLOSS=0.06
      ENDIF
C-----COMPONENT EFFICIENCIES
      PIMAX=0.97
      PIDMAX=0.97
      VCOEFF=0.97
    RETURN
    END
C======================================================================
C       INPUT DATA BASE SUBROUTINES                      ==============
C----------------------------------------------------------------------
      SUBROUTINE FLIGHT(FMACH,ALT,PSO,TSO)
      DIMENSION TT(0:50),PP(0:50)
      DATA TT/1.0000,0.9931,0.9863,0.9794,0.9725,0.9656,0.9583,0.9519,
     >0.9540,0.9381,0.9313,0.9244,0.9175,0.9107,0.9038,0.8969,0.8901,
     >0.8832,0.8764,0.8695,0.8626,0.8558,0.8489,0.8420,0.8352,0.8233,
```

```
>0.8215,0.8146,0.8077,0.8009,0.7940,0.7872,0.7803,0.7735,0.7666,
>0.7598,0.7529,0.7519,0.7519,0.7519,0.7519,0.7519,0.7519,0.7519,
>0.7519,0.7519,0.7519,0.7519,0.7519,0.7519,0.7519/
 DATA PP/1.0000,0.9644,0.9298,0.8963,0.8637,0.8321,0.8014,0.7717,
>0.7429,0.7149,0.6878,0.6616,0.6362,0.6115,0.5877,0.5646,0.5422,
>0.5206,0.4997,0.4795,0.4599,0.4410,0.4227,0.4051,0.3880,0.3716,
>0.3557,0.3404,0.3256,0.3113,0.2975,0.2843,0.2715,0.2592,0.2474,
>0.2360,0.2250,0.2145,0.2044,0.1949,0.1858,0.1771,0.1688,0.1609,
>0.1534,0.1462,0.1046,0.1329,0.1267,0.1208,0.1151/
   WRITE(6,*)'INPUT FLIGHT CONDITIONS'
   WRITE(6,*)'    INPUT FLIGHT MACH NUMBER:'
     READ(5,*)FMACH
   WRITE(6,*)'    INPUT ALTITUDE (0-50000 FT)'
     READ(5,*)ALT
     IALT=ALT/1000
     TSO=(459.67+59.0)*TT(IALT)
     PSO=14.696*PP(IALT)
 RETURN
 END
C
 SUBROUTINE DESLIM(TFUEL,FLHV,AMACH8A,HFUEL,PRLC,PRHC,PRAT,TPR,
>ICONFIG)
   COMMON CONFIG
   WRITE(6,*)'INPUT DESIGN/FUEL LIMITATIONS:'
   WRITE(6,*)'    TFUEL=(TEMP OF FUEL IN R)'
     READ(5,*)TFUEL
   WRITE(6,*)'    FLHV=(FUEL LOWER HEATING VALUE)'
     READ(5,*)FLHV
   WRITE(6,*)'    AMACH8=(EXIT MACH AT LPT)'
     READ(5,*)AMACH8A
   WRITE(6,*)'    PRLC='
     READ(5,*)PRLC
   WRITE(6,*)'    PRHC='
     READ(5,*)PRHC
     IF (ICONFIG.EQ.0) THEN
       TPR=1.0
     ENDIF
   PRAT=PRHC*PRLC*TPR
   HFUEL=TFUEL*0.5
 RETURN
 END
C
 SUBROUTINE COMPFOM(ETALC,ETAHC,ETAHT,ETALT,ETACC,ETAAB)
   WRITE(6,*)'INPUT COMPONENT FIGURES OF MERIT:(1=OWN 0=STD)'
   WRITE(6,*)'(STD: ETALC=0.86 ETAHC=0.88 ETAHT=0.87 ETALT=0.90
> ETACC=0.96 ETAAB=0.96)'
   READ(5,'(I)')ISUB2
   IF (ISUB2.EQ.0) THEN
      ETALC=0.86
      ETAHC=0.88
      ETAHT=0.87
      ETALT=0.90
      ETACC=0.96
      ETAAB=0.96
   ELSE
     WRITE(6,*)'    INPUT ETALC='
       READ(5,*)ETALC
     WRITE(6,*)'    INPUT ETAHC='
       READ(5,*)ETAHC
     WRITE(6,*)'    INPUT ETAHT='
```

```
          READ(5,*)ETAHT
        WRITE(6,*)'    INPUT ETALT='
          READ(5,*)ETALT
        WRITE(6,*)'    INPUT ETACC='
          READ(5,*)ETACC
        WRITE(6,*)'    INPUT ETAAB='
          READ(5,*)ETAAB
      ENDIF
    RETURN
    END
C
    SUBROUTINE TMAX(TMAXCC,TMAXAB,TMAXWR)
      WRITE(6,*)'INPUT MAX COMPONENT TEMPERATURES'
      WRITE(6,*)'    INPUT TMAXCC='
      WRITE(6,*)'       TMAXCC IS RESET TO TMAXWR IN ROTOR CONFIG'
        READ(5,*)TMAXCC
      WRITE(6,*)'    INPUT TMAXAB='
        READ(5,*)TMAXAB
      WRITE(6,*)'    INPUT TMAXWR='
      WRITE(6,*)'       TMAXWR IS RESET TO TREF IN ROTOR CONFIG'
        READ(5,*)TMAXWR
    RETURN
    END
C
    SUBROUTINE ACSYS(CTO,TMAXCC,XI,BETA)
      WRITE(6,*)'VEHICLE SYSTEM PARAMETERS'
C     WRITE(6,*)'    INPUT CTO= (TAKE-OFF POWER)'
C       READ(5,*)CTO
      WRITE(6,*)'    INPUT % COOLING OR BASE IT ON TMAXCC:'
      WRITE(6,*)'       (1=INPUT FOR XI     0=BASE XI ON TMAXCC)'
      READ(5,*)ICOOL1
      IF (ICOOL1.EQ.0) THEN
        IF (TMAXCC.GT.2400) THEN
          XI=2.0*(TMAXCC-2400.0)/16000.0
        ELSE
          XI=0.0
        ENDIF
      ELSE
       WRITE(6,*)'INPUT XI='
       READ(5,*)XI
      ENDIF
      WRITE(6,*)'    INPUT BETA= (BY-PASS RATIO)'
      READ(5,*)BETA
    RETURN
    END
C
    SUBROUTINE COMLOSS(BPLOSS,CCLOSS,ABLOSS)
      WRITE(6,*)'COMPONENT LOSSES (PERCENTS):(1=OWN 0=STD)'
      WRITE(6,*)'STD: BPLOSS=0.02 CCLOSS=0.03 ABLOSS=0.02 )'
        READ(5,'(I)')IQ5
      IF (IQ5.EQ.0) THEN
        BPLOSS=0.02
        CCLOSS=0.03
        ABLOSS=0.02
      ELSE
        WRITE(6,*)'    INPUT BPLOSS='
          READ(5,*)BPLOSS
        WRITE(6,*)'    INPUT CCLOSS='
          READ(5,*)CCLOSS
        WRITE(6,*)'    INPUT ABLOSS='
```

```
            READ(5,*)ABLOSS
          ENDIF
        RETURN
        END
C
        SUBROUTINE COMPETA(PIMAX,PIDMAX,VCOEFF)
          WRITE(6,*)'MISCELLANEOUS COMPONENT CHOICES:(1=OWN 0=STD)'
          WRITE(6,*)'(STD: PIMAX=0.97 PIDMAX=0.97 VCOEFF=0.97)'
          READ(5,'(I)')IQ6
          IF (IQ6.EQ.0) THEN
            PIMAX=0.97
            PIDMAX=0.97
            VCOEFF=0.97
          ELSE
            WRITE(6,*)'    INPUT PIMAX='
              READ(5,*)PIMAX
            WRITE(6,*)'    INPUT PIDMAX='
              READ(5,*)PIDMAX
            WRITE(6,*)'    INPUT VCOEFF= (NOZZLE EFF)'
              READ(5,*)VCOEFF
          ENDIF
        RETURN
        END
C
        SUBROUTINE WRSPECS(SPR,TPR,RREF,AMEX)
          WRITE(6,*)'WAVE ROTOR SPECIFICATIONS'
          WRITE(6,*)'    INPUT SPR='
            READ(5,*)SPR
          WRITE(6,*)'    INPUT TPR='
            READ(5,*)TPR
          WRITE(6,*)'    INPUT RREF=  kg/m^3'
            READ(5,*)RREF
          WRITE(6,*)'    INPUT AMEX='
            READ(6,*)AMEX
C------TO INPUT TCCIN CHANGE WRSPECS SUBR AND CALL-----------------
C         WRITE(6,*)'    INPUT TCCIN='
C           READ(5,*)TCCIN
        RETURN
        END
C
C ***     ***************************************************************
        SUBROUTINE INLET(PIN,TIN,FIN,PO,TO,FO,VO,FMACH,PIDMAX)
        COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
        COMMON/CONST/CJ,G
        HIN=ENTHFN(FIN,TIN)
        PHIIN=ENTRFN(FIN,TIN)
          GLOCIN=GLOC(FIN,TIN,HIN)
          PTIN=PIN*(1.+(GLOCIN-1.)/2.*FMACH**2.)**(GLOCIN/(GLOCIN-1.))
          TTIN=TIN*(1.+(GLOCIN-1.)/2.*FMACH**2.)
          PID=PIDMAX
          IF(FMACH.GE.1.0) PID=PIDMAX*(1.-0.075*(FMACH-1.)**1.35)
          DIFLOS=1.-PID
          PO=PTIN*(1-DIFLOS)
          TO=TTIN
          FO=0.0
        HO=HIN
        PHIO=PHIIN
        VO=FMACH*SQRT(GLOCIN*RGAS(FIN)*TIN*G)
        WRITE(9,10)
        WRITE(9,20)TIN,TO,PIN,PO,FIN,FO,VO
```

```
10    FORMAT(5X,'STATION IDENT: ENGINE INLET, STATIONS 0-1'/)
20    FORMAT(5X,'TI=',F5.0,2X,'TO=',F5.0,2X,'PI=',F5.1,2X,'PO=',
     >F5.1,2X,'FI=',F5.3,2X,'FO=',F4.3,2X,'VO=',F5.0/)
      RETURN
      END
C *** ****************************************************************
      SUBROUTINE COMPR(PR,PIN,TIN,PO,TO,TOIS,FIN,FO,IDC)
      COMMON/ETAS/ETALC,ETAHC,ETACC,ETAHT,ETALT,ETAAB,VCOEFF
      COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
      COMMON/WORK1/WORKLC,WORKHC
      COMMON/CONST/CJ,G
      IDC=IDC+1
        IF(IDC.EQ.1) GO TO 30
C *******COMPUTE HPC
      HIN=ENTHFN(FIN,TIN)
      PHIIN=ENTRFN(FIN,TIN)
      PHIOIS=PHIIN+RGAS(FIN)/CJ*ALOG(PR)
      CALL PHIFN(FIN,PHIOIS,TOIS)
      HOIS=ENTHFN(FIN,TOIS)
      HO=HIN+(HOIS-HIN)/ETAHC
      WORKHC=HO-HIN
      CALL HFN(FIN,HO,TO)
      PHIO=ENTRFN(FIN,TO)
      FO=FIN
      PO=PIN*PR
      GBAR=GAVG(FO,TIN,TOIS,PHIIN,PHIOIS)
      TEMPHC1=ALOG(TO/TIN)
      TEMPHC2=ALOG(PO/PIN)
        EHPC=(GBAR-1.0)/GBAR*TEMPHC2/TEMPHC1
C       WRITE(6,*)'TO=',TO,'TIN=',TIN,'PO=',PO,'PIN=',
C    >PIN,'ETAHC=',ETAHC,'GBAR=',GBAR
C      WRITE(6,*)'EHPC=',EHPC
        WRITE(9,10)
        WRITE(9,20)TIN,TO,PIN,PO,FIN,FO,ETAHC,GBAR,EHPC
10    FORMAT(5X,'STAION IDENT: HP COMPRESSOR, STATIONS 2-3'/)
20    FORMAT(5X,'TI=',F5.0,2X,'TO=',F5.0,2X,'PI=',F5.1,2X,'PO=',
     >F5.1,2X,'FI=',F4.3,2X,'FO=',F4.3/,5X,'ETAHC=',F6.4,2X,'GBAR=',
     >F6.4/,5X,'EHPC=',F6.4/)
      RETURN
C *****COMPUTE LPC
   30 HIN=ENTHFN(FIN,TIN)
      PHIIN=ENTRFN(FIN,TIN)
      PHIOIS=PHIIN+RGAS(FIN)/CJ*ALOG(PR)
      CALL PHIFN(FIN,PHIOIS,TOIS)
      HOIS=ENTHFN(FIN,TOIS)
      HO=HIN+(HOIS-HIN)/ETALC
      WORKLC=HO-HIN
      CALL HFN(FIN,HO,TO)
      PHIO=ENTRFN(FIN,TO)
      FO=FIN
      PO=PIN*PR
      GBAR=GAVG(FO,TIN,TOIS,PHIIN,PHIOIS)
C       WRITE(6,*)'TO=',TO,'TIN=',TIN,'PO=',PO,'PIN=',
C    >PIN,'ETALC=',ETALC,'GBAR=',GBAR
        TEMPLC1=ALOG(TO/TIN)
        TEMPLC2=ALOG(PO/PIN)
        ELPC=(GBAR-1.0)/GBAR*TEMPLC2/TEMPLC1
C       WRITE(6,*)'ELPC=',ELPC
        WRITE(9,40)
        WRITE(9,50)TIN,TO,PIN,PO,FIN,FO,ETALC,GBAR,ELPC
```

```
40    FORMAT(5X,'STATION IDENT: LP COMPRESSOR, STATIONS 1-2'/)
50    FORMAT(5X,'TI=',F5.0,2X,'TO=',F5.0,2X,'PI=',F5.1,2X,'PO=',
    >F5.1,2X,'FI=',F4.3,2X,'FO=',F4.3/,5X,'ETALC=',F6.4,2X,'GBAR=',
    >F6.4/,5X,'ELPC=',F6.4/)
     RETURN
     END
C ***    ************************************************************
C ***    ****WAVE ROTOR SUBROUTINE***********************************
C ***    ************************************************************
     SUBROUTINE ROTOR(PRAT1,PT1,PIN,PO,TIN,TO,PREF,RREF,TREF,FIN,FO,
    >SPR,TPP,AMEX,TCCIN,IDB1,CONFIG)
      COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
      COMMON/BURN/HFUEL,FLHV,TMAXCC,TMAXAB,TMAXWR
      COMMON/CONST/CJ,G
      DUMMY=0.
      GGUESS=1.3
      FO=FIN
      TSO=TMAXCC/(1.+(GGUESS-1.)/2.*AMEX**2.)
      KCT=0
10    KCT=KCT+1
      HSO=ENTHFN(FO,TSO)
      GLOCO=GLOC(FO,TSO,HSO)
      PO=PIN*TPR*(1.-CCLOSS)
      PSO=PO/(1.+(GLOCO-1.)/2.*AMEX**2.)**(GLOCO/(GLOCO-1.))
      PREF=PSO/SPR
C-------CONVERT DENSITY FROM kg/m^3 to lbm/ft^3-------------------
      TREF=PREF/RGAS(FO)/RREF*2306.6587
      STR=SPR**((GLOCO-1.)/GLOCO)
      TSO=STR*TREF
      TO=TSO*(1.+(GLOCO-1.)*AMEX**2./2.)
C     WRITE(6,*)'TREF=',TREF,'TMAXWR=',TMAXWR,'KCT=',KCT,'FO=',FO
      IF((ABS(TREF-TMAXWR).LT.5.0).OR.(KCT.GT.5)) GOTO 20
      TMAXWR=TREF
C-------COMPUTE TEMP INTO THE BURNER (TCCIN)-----------------------
        TCCIN=TIN*(PRAT1*PT1/PIN)**((GLOCO-1.)/GLOCO)
C       WRITE(6,*)'PRAT1=',PRAT1,'   PT1=',PT1
C       WRITE(6,*)'TCCIN = ',TCCIN
      CALL BURNER(DUMMY,TCCIN,DUMMY,TMAXWR,FIN,FO,IDB1,KODE,1)
      GOTO 10
20    CONFIG=0
      IDB1=IDB1+1
C     WRITE(6,*)'IDB1=',IDB1
      WRITE(9,30)
      WRITE(9,40)TCCIN,TO,TREF,PIN,PO,PREF,FIN,FO,GLOCO,IDB1
30    FORMAT(5X,'STATION IDENT: WAVE ROTOR, STATIONS 3-4'/)
40    FORMAT(5X,'TCCIN=',F7.1,2X,'TO=',F7.1,2X,'TREF=',F7.1/,5X,
    >'PIN=',F6.1,2X,'PO=',F6.1,2X,'PREF=',F6.1/,5X,'FIN=',F6.4,2X,
    >'FO=',F6.4,2X,'GLOCO=',F5.3,2X,'IDB1=',I1/)
      RETURN
      END
C
C ***    ************************************************************
     SUBROUTINE BURNER(PIN,TIN,PO,TO,FIN,FO,IDB2,KODE,CONFIG)
     INTEGER CONFIG
     COMMON/ETAS/ETALC,ETAHC,ETACC,ETALT,ETAAB,VCOEFF
     COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
     COMMON/BURN/HFUEL,FLHV,TMAXCC,TMAXAB,TMAXWR
C-----CHECK COUNTER OPERATION
C----------------CONFIG=1          GOTO WAVE ROTOR CALC
C----------------       CONFIG RESET = 0 IN ROTOR SUBROUTINE
```

```
C----------------     IDB INCREASED BY 1 IN ROTOR SUBR
C----------------CONFIG=0 AND IDB=1 GOTO MAIN CC CALC
C----------------CONFIG=0 AND IDB>1 GOTO A/B CALC
      IF(CONFIG.EQ.1) GOTO 50
      IDB2=IDB2+1
C       WRITE(6,*)'CONFIG=',CONFIG,'  IDB2=',IDB2
      IF(IDB2.EQ.1) GOTO 10
      GOTO 20
C-----MAIN BURNER CALC---------------------------------------------
   10 TO=TMAXCC
      ETAB=ETACC
      DELTAF=(A(TO)-A(TIN)+FIN*(B(TO)-B(TIN)))/(HFUEL+ETAB*FLHV-B(TO))
      FO=DELTAF+FIN
      PO=PIN*(1.-CCLOSS)
      HIN=ENTHFN(FIN,TIN)
      HO=ENTHFN(FO,TO)
      PHIIN=ENTRFN(FIN,TIN)
      PHIO=ENTRFN(FO,TO)
        WRITE(9,60)
        WRITE(9,70)TIN,TO,PIN,PO,FIN,FO
      RETURN
C-----AFTERBURNER CALC---------------------------------------------
C-----A/B ON
   20   IF(KODE.EQ.0) GOTO 30
      TO=TMAXAB
      ETAB=ETAAB
      DELTAF=(A(TO)-A(TIN)+FIN*(B(TO)-B(TIN)))/(HFUEL+ETAB*FLHV-B(TO))
        GOTO 40
C-----A/B OFF
   30 TO=TIN
      DELTAF=0.
   40 FO=DELTAF+FIN
      PO=PIN*(1.-ABLOSS)
      HIN=ENTHFN(FIN,TIN)
      HO=ENTHFN(FO,TO)
      PHIIN=ENTRFN(FIN,TIN)
      PHIO=ENTRFN(FO,TO)
        WRITE(9,80)
        WRITE(9,70)TIN,TO,PIN,PO,FIN,FO
      RETURN
C-----WAVE ROTOR CALC----------------------------------------------
   50   TO=TMAXWR
        ETAB=ETACC
      DELTAF=(A(TO)-A(TIN)+FIN*(B(TO)-B(TIN)))/(HFUEL+ETAB*FLHV-B(TO))
        FO=FIN+DELTAF
   60   FORMAT(5X,'STATION IDENT: MAIN BURNER, STATIONS 3-4'/)
   70   FORMAT(5X,'TI=',F5.0,2X,'TO=',F5.0,2X,'PI=',F5.1,2X,'PO=',F5.1,
     >2X,'FI=',F4.3,2X,'FO=',F4.3/)
   80 FORMAT(5X,'STATION IDENT: AFTERBURNER, STATIONS 9-10'/)
      RETURN
      END
C ***   ***************************************************************
      SUBROUTINE MIXCP(FIN1,FIN2,PIN1,TIN1,PIN2,TIN2,FO,PO,TO,VAR,IDM)
      COMMON/ETAS/ETALC,ETAHC,ETACC,ETAHT,ETALT,ETAAB,VCOEFF
      COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
      IDM=IDM+1
      IF(IDM.EQ.1) GOTO 10
      ZETA=VAR
      GOTO 20
   10 ZETA=VAR/(1.-VAR)
```

```
   20 HIN1=ENTHFN(FIN1,TIN1)
      HIN2=ENTHFN(FIN2,TIN2)
      PHIIN1=ENTRFN(FIN1,TIN1)
      PHIIN2=ENTRFN(FIN2,TIN2)
      HO=(ZETA*(1.+FIN1)*HIN1+(1.+FIN2)*HIN2)/(ZETA*(1.+FIN1)+(1.+FIN2))
      FO=(ZETA*FIN1+FIN2)/(1.+ZETA)
      PO=PIN2
      CALL HFN(FO,HO,TO)
      PHIO=ENTRFN(FO,TO)
      IF(IDM.EQ.1) GOTO 30
      GOTO 40
   30   WRITE(9,50)
        WRITE(9,60)TIN1,TIN2,TO,PO,FIN1,FIN2,FO
      RETURN
   40   WRITE(9,70)
        WRITE(9,60)TIN1,TIN2,TO,PO,FIN1,FIN2,FO
   50   FORMAT(5X,'STATION IDENT: PRE LPT MIXING, STATIONS 3,5-6'/)
   60   FORMAT(5X,'T1=',F5.0,2X,'T2=',F5.0,2X,'TO=',F5.0,2X,'PO=',F5.1,
     >2X,'F1=',F4.3,2X,'F2=',F4.3,2X,'FO=',F4.3/)
   70   FORMAT(5X,'STATION IDENT: PRE NOZZLE MIXING, STATIONS 9,10-11'/)
      RETURN
      END
C ***   ******************************************************************
      SUBROUTINE TURBIN(PIN,TIN,PO,TO,TOIS,FIN,FO,HO,PRAT,PRLC,PRHT,PRLT
     >,XI,BETA,KASET,IDT)
      COMMON/ETAS/ETALC,ETAHC,ETACC,ETAHT,ETALT,ETAAB,VCOEFF
      COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
      COMMON/WORK1/WORKLC,WORKHC
      COMMON/CONST/CJ,G
      IDT=IDT+1
      HIN=ENTHFN(FIN,TIN)
      PHIIN=ENTRFN(FIN,TIN)
      IF(IDT.EQ.1) GOTO 10
      IF(KASET.EQ.1) GOTO 20
      GOTO 30
C-----HPT CALCULATION------------------------------------------
   10 HO=HIN-WORKHC/((1.-XI)*(1.+FIN))
      CALL HFN(FIN,HO,TO)
      PHIO=ENTRFN(FIN,TO)
      HOIS=HIN-(HIN-HO)/ETAHT
      CALL HFN(FIN,HOIS,TOIS)
      PHIOIS=ENTRFN(FIN,TOIS)
      PRHT=1./EXP(CJ/RGAS(FIN)*(PHIOIS-PHIIN))
      FO=FIN
      PO=PIN/PRHT
      GBAR=GAVG(FO,TIN,TOIS,PHIIN,PHIOIS)
        TEMPHT1=ALOG(PO/PIN)
        TEMPHT2=ALOG(TO/TIN)
        EHPT=GBAR/(GBAR-1.)*TEMPHT2/TEMPHT1
C       WRITE(6,*)'TO=',TO,'TIN=',TIN,'PO=',PO,
C    >'PIN=',PIN,'GBAR=',GBAR,'ETAHT=',ETAHT
C       WRITE(6,*)'EHPT=',EHPT
        WRITE(9,40)
        WRITE(9,50)TIN,TO,PIN,PO,FIN,FO,ETAHT,GBAR,EHPT
      RETURN
C-----CALCULATE OPTIMUM BYPASS RATIO
   20 PRLT=1./((1.-BPLOSS)/(1.-CCLOSS)*PRLC/PRAT*PRHT)
      PHIOIS=ENTRFN(FIN,TIN)+RGAS(FIN)/CJ*ALOG(1./PRLT)
      CALL PHIFN(FIN,PHIOIS,TOIS)
      HOIS=ENTHFN(FIN,TOIS)
```

```
      HO=HIN-ETALT*(HIN-HOIS)
      CALL HFN(FIN,HO,TO)
      PHIO=ENTRFN(FIN,TO)
      BETA=(1.+FIN)*(HIN-HO)/(WORKLC)-1.
C        WRITE(6,*)'TURBIN BETA=',BETA
      FO=FIN
      PO=PIN/PRLT
      GBAR=GAVG(FO,TIN,TOIS,PHIIN,PHIOIS)
        TEMPLT1=ALOG(PO/PIN)
        TEMPLT2=ALOG(TO/TIN)
        ELPT=GBAR/(GBAR-1.)*TEMPLT2/TEMPLT1
        WRITE(9,60)
        WRITE(9,50)TIN,TO,PIN,PO,FIN,FO,ETALT,GBAR,ELPT
      RETURN
C-----LPT CALCULATION BYPASS PRESCRIBED------------------------
   30 HO=HIN-(1.+BETA)*(WORKLC)/(1.+FIN)
      CALL HFN(FIN,HO,TO)
      PHIO=ENTRFN(FIN,TO)
      HOIS=HIN+(HO-HIN)/ETALT
      CALL HFN(FIN,HOIS,TOIS)
      PHIOIS=ENTRFN(FIN,TOIS)
      PRLT=1./EXP(CJ/RGAS(FIN)*(PHIOIS-PHIIN))
      FO=FIN
      PO=PIN/PRLT
      GBAR=GAVG(FO,TIN,TOIS,PHIIN,PHIOIS)
        TEMPLT1=ALOG(PO/PIN)
        TEMPLT2=ALOG(TO/TIN)
        ELPT=GBAR/(GBAR-1.)*TEMPLT2/TEMPLT1
C        WRITE(6,*)'TO=',TO,'TIN=',TIN,'PO=',PO,'PIN=',PIN,
C     >'GBAR=',GBAR,'ETALT=',ETALT
C        WRITE(6,*)'ELPT=',ELPT
        WRITE(9,60)
        WRITE(9,50)TIN,TO,PIN,PO,FIN,FO,ETALT,GBAR,ELPT
   40   FORMAT(5X,'STATION IDENT: HP TURBINE, STATIONS 4-5'/)
   50   FORMAT(5X,'TI=',F5.0,2X,'TO=',F5.0,2X,'PI=',F5.1,2X,'PO=',
     >F5.1,2X,'FI=',F4.3,2X,'FO=',F4.3/,5X,'ETA=',F6.4,2X,'GBAR=',
     >F6.4,5X,'ET=',F6.4/)
   60   FORMAT(5X,'STATION IDENT: LP TURBINE, STATIONS 6-8'/)
      RETURN
      END
C ***   ***********************************************************
      SUBROUTINE DUCT(PIN,TIN,FIN,PO,TO,FO,HO)
      COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
      PO=PIN*(1.-BPLOSS)
      TO=TIN
      FO=0.0
      HIN=ENTHFN(FIN,TIN)
      PHIIN=ENTRFN(FIN,TIN)
      HO=ENTHFN(FO,TO)
      PHIO=ENTRFN(FO,TO)
        WRITE(9,10)
        WRITE(9,20)TIN,TO,PIN,PO,FIN,FO
   10   FORMAT(5X,'STATION IDENT: BYPASS DUCT, STATIONS 2-7'/)
   20   FORMAT(5X,'TI=',F5.0,2X,'TO=',F5.0,2X,'PI=',F5.1,2X,'PO=',
     >F5.1,2X,'FI=',F4.3,2X,'FO=',F4.3/)
      RETURN
      END
C ***   ***************************************************************
C
      SUBROUTINE MIXCA(PIN2,PIN1,TIN2,TIN1,HIN2,HIN1,FIN2,FIN1,BETA,CMIN
```

```
     >1,PIMAX,TO,FO,PO,CMO)
      COMMON/CONST/CJ,G
C*****************************************STATEMENT FUNCTION DEFINITIONS
      PHI1FN(GAMMA,AMACH)=AMACH**2.*(1.+(GAMMA-1.)/2.*AMACH**2.)/((1.+GA
     >MMA*AMACH**2.)**2.)
      PSI1FN(G,GAMMA,R,AMACH)=(AMACH*SQRT(G*GAMMA/R))/(1.+(GAMMA-1.)/2.*
     >AMACH**2.)**((GAMMA+1.)/(2.*(GAMMA-1.)))
C****************************************SOLVE FOR FAN STREAM MACH NUMBER
C     CHECK IF BYPASS GREATER THAN 0.0
        IF (BETA.GT.0.0) THEN
                GOTO 5
        ELSE
                TO=TIN1
                PO=PIN1*PIMAX
                HO=HIN1
                FO=FIN1
                CMO=CMIN1
                AI2AI1=0.0
        ENDIF
        GO TO 9
C**************************************
    5   GLOCI2=GLOC(FIN2,TIN2,HIN2)
C       WRITE(6,*)'AMACH8=',CMIN1
        GLOCI1=GLOC(FIN1,TIN1,HIN1)
        FLAG1=(PIN2/PIN1*(1.+(GLOCI1-1.)/2.*CMIN1**2.)**(GLOCI1/(GLOCI
     >1-1.)))**((GLOCI2-1.)/GLOCI2)
C       WRITE(6,*)'FLAG1=',FLAG1
        IF (BETA.GT.0.0) THEN
          IF(FLAG1.LE.1.0) GOTO 60
        ENDIF
      CMIN2=SQRT(2./(GLOCI2-1.)*((PIN2/PIN1*(1.+(GLOCI1-1.)/2.*CMIN1**2.
     >)**(GLOCI1/(GLOCI1-1.)))**((GLOCI2-1.)/GLOCI2)-1.))
      IF(CMIN2.GE.0.8) GOTO 40
C**********************************SOLVE FOR MIXER INLET AREA RATIO
      AI2AI1=BETA/(1.+FIN1)*CMIN1/CMIN2*SQRT((GLOCI1*RGAS(FIN2)*TIN2*(1.
     >+(GLOCI1-1.)/2.*CMIN1**2.))/(GLOCI2*RGAS(FIN1)*TIN1*(1.+(GLOCI2-1.
     >)/2.*CMIN2**2.)))
C**************************************SOLVE FOR MIXER TEMPERATURE RATIO
      RIN2=RGAS(FIN2)
      RIN1=RGAS(FIN1)
      CPIN2=CP(FIN2,TIN2)
      CPIN1=CP(FIN1,TIN1)
      BETAPR=BETA/(1.+FIN1)
      CPO=(CPIN1+BETAPR*CPIN2)/(1.+BETAPR)
      TO=TIN1*(CPIN1/CPO*(1.+BETAPR*CPIN2*TIN2/(CPIN1*TIN1))/(1.+BETAPR)
     >)
      TAUMIX=TO/TIN1
C**********************************SOLVE FOR MIXER EXIT MACH NUMBER
      FO=(BETA*FIN2+FIN1)/(1.+BETA)
      HO=ENTHFN(FO,TO)
      RO=RGAS(FO)
      GLOCO=GLOC(FO,TO,HO)
      PHIIN2=PHI1FN(GLOCI2,CMIN2)
      PHIIN1=PHI1FN(GLOCI1,CMIN1)
      PHIO=(((1.+BETAPR)/(SQRT(1./PHIIN1)+BETAPR*SQRT(1./PHIIN2)*SQRT(RI
     >N2*TIN2*GLOCI1/(RIN1*TIN1*GLOCI2)))))**2.)*GLOCI1*RO*TAUMIX/(GLOCO*
     >RIN1)
      CMO=SQRT(2.*PHIO/((1.-2.*GLOCO*PHIO)+SQRT(1.-2.*(GLOCO+1.)*PHIO)))
C*****************************************SOLVE FOR MIXER PRESSURE RATIO
      POI=PIN1*(1.+BETAPR)*(1./(1.+AI2AI1))*SQRT(TAUMIX)*PSI1FN(G,GLOCI1
```

```
     >,RIN1,CMIN1)/PSI1FN(G,GLOCO,RO,CMO)
        A1=PSI1FN(G,GLOCI1,RIN1,CMIN1)
        A2=PSI1FN(G,GLOCO,RO,CMO)
      PIMIXI=POI/PIN1
      PIMIX=PIMIXI*PIMAX
      PO=PIMIX*PIN1
    9   WRITE(9,10)
        WRITE(9,20)TIN1,TIN2,TO,PIN1,PIN2,PO
        WRITE(9,30)FIN1,FIN2,FO,CMIN1,CMIN2,CMO,AI2AI1
      RETURN
   10 FORMAT(5X,'STATION IDENT: CONST. AREA MIXER, STATIONS 7,8-9'/)
   20 FORMAT(5X,'T1=',F5.0,2X,'T2=',F5.0,2X,'TO=',F5.0,2X,'P1=',F5.1,2X,
     >'P2=',F5.1,2X,'PO=',F5.1/)
   30 FORMAT(5X,'F1=',F4.3,2X,'F2=',F4.3,2X,'FO=',F4.3,2X,'M1=',F4.2,2X,
     >'M2=',F4.2,2X,'MO=',F4.2,2X,'ARAT=',F4.2/)
C
   40 WRITE(6,50)
   50 FORMAT(5X,'FAN STREAM MACH NO IS GREATER THAN/EQUAL TO 0.8')
        RETURN
C
   60 WRITE(6,70)
   70 FORMAT(5X,'FAN STREAM MACH NO IS LESS THAN/EQUAL TO 0.0')
        RETURN
      END
C *** ******************************************************************
C
      SUBROUTINE NOZZLE(PIN,TIN,PO,TO,TOIS,FIN,PRLC,VEXIT,EXMACH,PSO,
     >IFLAG)
      COMMON/ETAS/ETALC,ETAHC,ETALC,ETAHT,ETALT,ETAAB,VCOEFF
      COMMON/LOSSES/DIFLOS,CCLOSS,ABLOSS,BPLOSS
      COMMON/CONST/CJ,G
C      PRN=1./((1.-DIFLOS)*(1.-BPLOSS)*(1.-ABLOSS)*(PRLC))
      PRN=PSO/PIN
      HIN=ENTHFN(FIN,TIN)
      PHIIN=ENTRFN(FIN,TIN)
      GLOCO=GLOC(FIN,TIN,HIN)
      PRN=PSO/PIN
        IF(PRN.GE.1.0) GOTO 30
C        write(6,*)'GLOCO=',GLOCO
C        write(6,*)'PSO=',PSO
C       write(6,*)'PRN=',PRN
      PHIOIS=PHIIN+RGAS(FIN)/CJ*ALOG(PRN)
      CALL PHIFN(FIN,PHIOIS,TOIS)
      HOIS=ENTHFN(FIN,TOIS)
      PO=PIN*PRN
      ETAN=VCOEFF**2.
      HO=HIN-ETAN*(HIN-HOIS)
      CALL HFN(FIN,HO,TO)
      VEXIT=SQRT(2.*G*CJ*(HIN-HO))
      AEXIT=SQRT(G*RGAS(FIN)*TO/(1.-RGAS(FIN)/CJ*TO/HO))
      EXMACH=VEXIT/AEXIT
        WRITE(9,10)
        WRITE(9,20)TIN,TO,PIN,PO,FIN,VEXIT,EXMACH
        WRITE(9,25)GLOCO,PRN,AEXIT
      GOTO 40
   10 FORMAT(5X,'STATION IDENT: EXHAUST NOZZLE, STATIONS 11-12'/)
   20 FORMAT(5X,'TI=',F5.0,2X,'TO=',F5.0,2X,'PI=',F5.1,2X,'PO=',F5.1,
     >2X,'FI=',F4.3,2X,'VO=',F5.0,2X,'MO=',F4.2)
   25 FORMAT(5X,'GLOCO=',F6.4,2X,'PRN=',F6.5,2X,'AEXIT=',F6.1/)
   30 IFLAG=1
```

```
   40 RETURN
      END
C *** ******************************************************************
C
      SUBROUTINE PERF(AMACH8,FIN,VO,VEXIT,BETA,SPTHR,SFC,EXMACH)
      COMMON/CONST/CJ,G
      SPTHR=((1.+FIN)*VEXIT-VO)/G
      SFC=FIN*3600./SPTHR
      WRITE(9,10)
      WRITE(9,20)AMACH8,SPTHR,SFC,BETA,EXMACH
        WRITE(9,25)
   10 FORMAT(5X,'PERFORMANCE PARAMETERS'/)
   20 FORMAT(5X,'AMACH8=',F6.3/,5X,'SP.THRUST =',F6.2,2X,'SFC = ',F5.3
     >,4X,'BETA= ',F6.3,2X,'EXMACH=',F6.3)
   25 FORMAT(5X,'(LBF/LBM/S)',8X,'(LBM/HR/LBF)'/)
      RETURN
      END
C *** ******************************************************************
C
      SUBROUTINE HFN(FRATIO,HEX,TEMPEX)
      DUMMY=1.
      ITERNO=0
      IHEX=IFIX(HEX/100.)*100
      HONE=FLOAT(IHEX)
      HTWO=HONE+100.
      CALL TEMP(HONE,HTWO,DUMMY,DUMMY,TEMP1,TEMP2)
   10 ITERNO=ITERNO+1
      IF(ITERNO.GT.50) GOTO 20
      HONE=ENTHFN(FRATIO,TEMP1)
      HTWO=ENTHFN(FRATIO,TEMP2)
      SLOPE=(HTWO-HONE)/(TEMP2-TEMP1)
      TEMP1=TEMP1+(HEX-HONE)/SLOPE
      HONE=ENTHFN(FRATIO,TEMP1)
      IF(ABS(HONE-HEX).GT.0.002) GOTO 10
      TEMPEX=TEMP1
      GOTO 30
   20 WRITE(6,40)
   40 FORMAT(5X,'SECANT METHOD ITERATION DOES NOT CONVERGE'/)
   30 RETURN
      END
C *** ******************************************************************
C
      SUBROUTINE PHIFN(FRATIO,PHIEX,TEMPEX)
      DUMMY=1.
      ITERNO=0
      IPHIEX=IFIX(PHIEX*10.)
      PHIONE=FLOAT(IPHIEX)*0.1
      PHITWO=PHIONE+0.1
      CALL TEMP(DUMMY,DUMMY,PHIONE,PHITWO,TEMP1,TEMP2)
   10 ITERNO=ITERNO+1
      IF(ITERNO.GT.55) GOTO 20
      PHIONE=ENTRFN(FRATIO,TEMP1)
      PHITWO=ENTRFN(FRATIO,TEMP2)
      SLOPE=(PHITWO-PHIONE)/(TEMP2-TEMP1)
      TEMP1=TEMP1+(PHIEX-PHIONE)/SLOPE
      PHIONE=ENTRFN(FRATIO,TEMP1)
      IF(ABS(PHIONE-PHIEX).GT.0.000003) GOTO 10
      TEMPEX=TEMP1
      GOTO 40
   20 WRITE(6,35)
```

```
   35 FORMAT(5X,'SECANT METHOD ITERATION DOES NOT CONVERGE'/)
   40 RETURN
      END
C *** *****************************************************************
C        FUNCTION STATEMENTS
C *** *****************************************************************
      FUNCTION ENTHFN(FRATIO,TEMP)
        IF(TEMP.LT.0.0) GOTO 10
      C1=0.24062
      C2=0.017724E-03
      C3=0.038056E-06
      C4=0.012662E-09
      C5=0.0013012E-12
      D1=0.22091
      D2=0.51822E-03
      D3=0.19462E-06
      D4=0.045089E-09
      D5=0.0043275E-12
      ENTHFN=(1./(1.+FRATIO))*(C1*TEMP-C2/2.*TEMP**2.+C3/3.*TEMP**3.-C4/
     >4.*TEMP**4.+C5/5.*TEMP**5.)+(FRATIO/(1.+FRATIO))*(D1*TEMP+D2/2.*TE
     >MP**2.-D3/3.*TEMP**3.+D4/4.*TEMP**4.-D5/5.*TEMP**5.)
        GOTO 25
   10 WRITE(6,20)TEMP
   20 FORMAT(5X,'NEGATIVE TEMP IN ENTHFN ENCOUNTERED=',F6.1/)
   25   CONTINUE
      RETURN
      END
C
      FUNCTION ENTRFN(FRATIO,TEMP)
        IF(TEMP.LE.0.0) GOTO 10
      C1=0.24062
      C2=0.017724E-03
      C3=0.038056E-06
      C4=0.012662E-09
      C5=0.0013012E-12
      D1=0.22091
      D2=0.51822E-03
      D3=0.19462E-06
      D4=0.045089E-09
      D5=0.0043275E-12
      ENTRFN=(1./(1.+FRATIO))*(C1*ALOG(TEMP)-C2*TEMP+C3*0.5*TEMP**2.-C4/
     >3.*TEMP**3.+C5/4.*TEMP**4.)+(FRATIO/(1.+FRATIO))*(D1*ALOG(TEMP)+D2
     >*TEMP-D3*0.5*TEMP**2.+D4/3.*TEMP**3.-D5/4.*TEMP**4.)
      RETURN
   10 WRITE(6,20)TEMP
   20 FORMAT(5X,'NEGATIVE/ZERO TEMP IN ENTRFN ENCOUNTERED =',F6.1)
      RETURN
      END
C
      FUNCTION A(T)
      C1=.24062
      C2=1.7724E-5
      C3=3.8056E-8
      C4=1.2662E-11
      C5=1.3012E-15
      D1=.22091
      D2=5.1822E-4
      D3=1.9462E-7
      D4=4.5089E-11
      D5=4.3275E-15
```

```
      A=C1*T-C2/2.*T**2.+C3/3.*T**3.-C4/4.*T**4.+C5/5.*T**5.
      RETURN
      END
C
      FUNCTION B(T)
      C1=.24062
      C2=1.7724E-5
      C3=3.8056E-8
      C4=1.2662E-11
      C5=1.3012E-15
      D1=.22091
      D2=5.1822E-4
      D3=1.9462E-7
      D4=4.5089E-11
      D5=4.3275E-15
      B=D1*T+D2/2.*T**2.-D3/3.*T**3.+D4/4.*T**4.-D5/5.*T**5.
      RETURN
      END
C
      FUNCTION CP(FRATIO,TEMP)
      C1=0.24062
      C2=0.017724E-03
      C3=0.038056E-06
      C4=0.012662E-09
      C5=0.0013012E-12
      D1=0.22091
      D2=0.51822E-03
      D3=0.19462E-06
      D4=0.045089E-09
      D5=0.0043275E-12
      CP=(1./(1.+FRATIO))*(C1-C2*TEMP+C3*TEMP**2.-C4*TEMP**3.+C5*TEMP**4
     >.)+(FRATIO/(1.+FRATIO))*(D1+D2*TEMP-D3*TEMP**2.+D4*TEMP**3.-D5*TEM
     >P**4.)
      RETURN
      END
C
      FUNCTION RGAS(FRAT)
      WMOL=(1.+FRAT)/(0.034522+0.035648*FRAT)
      RGAS=1545.43/WMOL
      RETURN
      END
C
      FUNCTION GAVG(FRAT,TIN,TOIS,PHIIN,PHIOIS)
      COMMON/CONST/CJ,G
C      WRITE(6,5)FRAT,TIN,TOIS,PHIIN,PHIOIS,CJ
C    5 FORMAT(5X,F3.1,5X,2F8.1,5X,2F7.2,5X,F6.2)
      GAVG=1./(1.-(RGAS(FRAT)/CJ)*(ALOG(TOIS/TIN)/(PHIOIS-PHIIN)))
      RETURN
      END
C
      FUNCTION GLOC(FLOC,TLOC,HLOC)
      COMMON/CONST/CJ,G
      GLOC=1./(1.-RGAS(FLOC)/CJ*TLOC/HLOC)
      RETURN
      END
C
C
      SUBROUTINE TEMP(HONE,HTWO,PHIONE,PHITWO,TEMP1,TEMP2)
      IF((HONE.EQ.0.00).OR.(PHIONE.LE.1.3)) GOTO 10
      IF((HONE.EQ.100.).OR.(PHIONE.EQ.1.4)) GOTO 20
```

```
      IF((HONE.EQ.200.).OR.(PHIONE.EQ.1.5)) GOTO 30
      IF((HONE.EQ.300.).OR.(PHIONE.EQ.1.6)) GOTO 40
      IF((HONE.EQ.400.).OR.(PHIONE.EQ.1.7)) GOTO 50
      IF((HONE.EQ.500.).OR.(PHIONE.EQ.1.8)) GOTO 60
      IF((HONE.GE.600.).OR.(PHIONE.GE.1.9)) GOTO 70
      WRITE(6,1)HONE,PHIONE
    1 FORMAT(5X,'NOTHING MATCHED:HONE=',F13.9,5X,'PHIONE=',F13.9/)
      STOP
   10 TEMP1=0.
      TEMP2=450.
      RETURN
   20 TEMP1=300.
      TEMP2=800.
      RETURN
   30 TEMP1=500.
      TEMP2=1250.
      RETURN
   40 TEMP1=750.
      TEMP2=1600.
      RETURN
   50 TEMP1=1100.
      TEMP2=2000.
      RETURN
   60 TEMP1=1500.
      TEMP2=2400.
      RETURN
   70 TEMP1=2200.
      TEMP2=3300.
      RETURN
      END
C *** ******************************************************************
C        PLOT SUBROUTINES USING 'DISSPLA' SOFTWARE - FACILITY DEPENDENT
C *** ******************************************************************
C       SUBROUTINE PLOT(XARAY,YARAY,ZARAY,BPR,JJ,I1)
C       DIMENSION XARAY(I1),YARAY(I1),ZARAY(I1),BPR(I1)
C       REAL XARAY,YARAY,ZARAY,BPR
C       INTEGER I1,JJ
C       CALL PHYSOR(0.5,0.5)
C       CALL AREA2D(10.,7.5)
C       CALL FRAME
C       CALL GRAF(0.,'SCALE',140.,0.,'SCALE',100.)
C       CALL GRAF(50.,'SCALE',120.,0.6,'SCALE',1.6)
C       CALL XNAME('SP.THRUST',9)
C       CALL YNAME('TSFC',4)
C       GOTO(10,20,30,40,50,10,20,30,40,50),JJ
C    10 CALL SETCLR('YELLOW')
C       CALL MARKER(15)
C       GOTO 60
C    20 CALL SETCLR('CYAN')
C       CALL MARKER(18)
C       GOTO 60
C    30 CALL SETCLR('MAGENTA')
C       CALL MARKER(5)
C       GOTO 60
C    40 CALL SETCLR('GREEN')
C       CALL MARKER(9)
C       GOTO 60
C    50 CALL SETCLR('RED')
C       CALL MARKER(2)
C    60 CALL CURVE(XARAY,YARAY,I1,+1)
```

```
C       CALL ENDGR(0)
C       RETURN
C       END
C-----------------------------------------------------------------------
C        PLOT ROUTINE USING GRAFKIT SOFTWARE
C------------------------------------------------------------------------
      SUBROUTINE GKSPLOT1(X1A,Y1A,Y2A,IAN4,ICP1)
C-----PLOT OF SPECIFIC THRUST AND SFC VS INPUT PARAMETER
C-----CHOICES OF INPUT PARAMETERS ARE:
C-----           1. BYPASS RATIO
C-----           2. FAN PRESSURE RATIO
C-----           3. COMPRESSOR PRESSURE RATIO
C-----           4. LPT EXIT MACH NUMBER
C
      REAL X1A(50),Y1A(50),Y2A(50)
      CHARACTER X1LABEL*6,Y1LABEL*33,Y2LABEL*33
      CHARACTER FIG1*7,NUMB1*3,TOPLBL*10
C------------------------------------------------------------
C-----SPECIFY X,Y LEFT AND Y RIGHT LABELS------------------
C
         Y1LABEL='A: Specific Thrust    lbf/(lbm/s)'
         Y2LABEL='B: SFC   (lbm fuel/hr)/lbf thrust'
C-----CHOOSE THE X PARAMETER TO PLOT
         IF (IAN4.EQ.1) THEN
           X1LABEL=' BETA '
          ELSEIF (IAN4.EQ.2) THEN
           X1LABEL=' PRLC '
          ELSEIF (IAN4.EQ.3) THEN
           X1LABEL=' PRHC '
          ELSEIF (IAN4.EQ.4) THEN
           X1LABEL='AMACH8'
         ENDIF
C-----
        WRITE(6,*)'INPUT: '
        WRITE(6,*)'    0 - AUTOMATICALLY DETERMINE X1MIN AND X1MAX'
        WRITE(6,*)'    1 - INPUT X1MIN AND X1MAX'
         READ(5,*)IANS5
          IF (IANS5.EQ.1) THEN
             WRITE(6,*)' INPUT X1MIN =:'
             READ(5,*)X1MIN
             WRITE(6,*)' INPUT X1MAX =:'
             READ(5,*)X1MAX
          ELSE
C-----      AUTOMATICALLY DETERMINE THE MIN AND MAX X VALUE
            X1MIN=X1A(1)
            X1MAX=X1A(1)
            DO J=1,ICP1
              X1TEMP=X1A(J)
              X2TEMP=X1A(J)
              IF (X1MIN.GT.X1TEMP) THEN
                 X1MIN=X1TEMP
              ENDIF
              IF (X1MAX.LT.X2TEMP) THEN
                 X1MAX=X2TEMP
              ENDIF
            END DO
          ENDIF
C
C        WRITE(6,*)'X1MIN=',X1MIN,'X1MAX=',X1MAX
C
```

```
C-----INPUT THE NUMBER OF THE FIGURE TO PLOT (FIG 1)---
        WRITE(6,*)'INPUT THE FIGURE NUMBER (ie 1 OR 2...)'
          READ(5,'(A9)')NUMB1
C       WRITE(6,*)'NUMB1=',NUMB1
        FIG1='FIGURE '
        TOPLBL=FIG1//NUMB1
C       WRITE(6,'(1X,A11)')'TOP LABEL =',TOPLBL
C-------------------------------------------------------
C-----CHECK THE DATA INPUT
C      DO I1=1,ICP1
C       WRITE(6,*)I1,X1A(I1),Y1A(I1),Y2A(I1)
C      END DO
C-----CONSTRUCT THE PLOT
C
      WRITE(6,*)'STANDBY FOR PLOT'
C-----OPEN GKS, OPEN AND ACTIVATE WORKSTATION.
        CALL OPNGKS(1,2,1)
C
C     SET UP THE COLOR TABLE
        CALL GSCR(1,0,0.00,0.00,0.50)
        CALL GSCR(1,1,1.00,1.00,0.00)
        CALL GSCR(1,2,0.00,1.00,0.50)
        CALL GSCR(1,3,0.00,1.00,1.00)
        CALL GSCR(1,4,1.00,0.00,0.00)
        CALL GSCR(1,5,1.00,0.00,1.00)
        CALL GSCR(1,6,1.00,0.00,1.00)
        CALL GSCR(1,7,1.00,1.00,1.00)
        CALL GSCR(1,8,0.00,0.00,0.00)
C
C-----SET VIEWPOINT AND DEFINE THE RANGE OF USER'S DATA TO BE USED
        CALL AGSETF ('GRID/LEFT.',    .15)
        CALL AGSETF ('GRID/RIGHT.',   .85)
        CALL AGSETF ('GRID/TOP.',     .85)
        CALL AGSETF ('GRID/BOTTOM.', .15)
C-----SET UP FOR FIRST PLOT
C----- DEFINE THE TOP, BOTTOM, LEFT AND RIGHT LABELS
C----- PUT BLANKS IN FIRST CURVE LABELS
        CALL AGSETC ('LABEL/NAME.',   'T')
        CALL AGSETI ('LINE/NUMBER.', 100)
        CALL AGSETC ('LINE/TEXT.',    ' ')
C
        CALL AGSETC ('LABEL/NAME.',   'B')
        CALL AGSETI ('LINE/NUMBER.',-100)
        CALL AGSETC ('LINE/TEXT.',    ' ')
C
        CALL AGSETC ('LABEL/NAME.',   'L')
        CALL AGSETI ('LINE/NUMBER.', 100)
        CALL AGSETC ('LINE/TEXT.',    ' ')

        CALL AGSETC ('LABEL/NAME.',   'R')
        CALL AGSETI ('LINE/NUMBER.',-100)
        CALL AGSETC ('LINE/TEXT.',    ' ')
C-----TURN OFF RIGHT AXIS
        CALL AGSETF ('AXIS/RIGHT/CONTROL.',0.0)
C-----SET YMIN AND YMAX FOR FIRST PLOT
        CALL AGSETF ('X/MINIMUM.', X1MIN)
        CALL AGSETF ('X/MAXIMUM.', X1MAX)
        CALL AGSETF ('Y/MINIMUM.',   0.0)
        CALL AGSETF ('Y/MAXIMUM.', 120.0)
C------
```

```
        CALL AGSTUP (X1A,1,1,ICP1,1,Y1A,1,1,ICP1,1)
        CALL AGBACK
        CALL GSPLCI (7)
        CALL DASHDC ('$$$$$$$$$$$''''''''''$$$$$$$$$$',3,1)
C------PLOT CURVE 1: X1A VS Y1A
        CALL AGCURV (X1A,1,Y1A,1,ICP1,-1)
C
C------------------------
C-----SET UP FOR SECOND PLOT
C----- DEFINE THE TOP, BOTTOM, LEFT AND RIGHT LABELS
C----- THE LARGER LINE NUMBER WRITES ABOVE PREVIOUS LABELS
        CALL AGSETC ('LABEL/NAME.',  'T')
        CALL AGSETI ('LINE/NUMBER.', 100)
        CALL AGSETC ('LINE/TEXT.', TOPLBL)
C
        CALL AGSETC ('LABEL/NAME.',  'B')
        CALL AGSETI ('LINE/NUMBER.',-110)
        CALL AGSETC ('LINE/TEXT.',X1LABEL)
C
        CALL AGSETC ('LABEL/NAME.',  'L')
        CALL AGSETI ('LINE/NUMBER.', 110)
        CALL AGSETC ('LINE/TEXT.',Y1LABEL)

        CALL AGSETC ('LABEL/NAME.',  'R')
        CALL AGSETI ('LINE/NUMBER.',-100)
        CALL AGSETC ('LINE/TEXT.',Y2LABEL)
C     TURN OFF THE TOP, BOTTOM AND LEFT AXES
        CALL AGSETF ('AXIS/TOP/CONTROL.',0.0)
        CALL AGSETF ('AXIS/BOTTOM/CONTROL.',0.0)
        CALL AGSETF ('AXIS/LEFT/CONTROL.',0.0)
C-----TURN ON THE RIGHT AXIS
        CALL AGSETF ('AXIS/RIGHT/CONTROL.',1.0)
C-----CHANGE THE YMIN AND YMAX
        CALL AGSETF ('Y/MINIMUM.', 0.0)
        CALL AGSETF ('Y/MAXIMUM.', 1.5)
C-----TURN ON THE RIGHT AXIS NUMERICS
        CALL AGSETF ('RIGHT/NUMERIC/TYPE.',1.E6)
C
        CALL AGSTUP (X1A,1,1,ICP1,1,Y2A,1,1,ICP1,1)
        CALL AGBACK
        CALL GSPLCI (4)
        CALL DASHDC ('$$SFC$''''',8,1)
C------PLOT CURVE 2: X1A VS Y2A
        CALL AGCURV (X1A,1,Y2A,1,ICP1,-2)
C-----CALL FRAME TO ADVANCE THE FRAME
C-----CLOSE GKS, CLOSE AND DEACTIVATE WORKSTATION
        CALL FRAME
        CALL CLSGKS
C----------------------
      END
```

APPENDIX B

PROGRAM WRCOMP

A. PROCEDURES

1. WRCOMP

   a. Log on to the VAX

   b. At the DCL prompt type:

```
$run WRCOMP1A..........runs WRCOMP program
          ..........select options from screen
```

2. INTERFACING WITH ENGINE

   a. Run ENGINE program in either configuration.

   b. $type ENGINE1.DAT or
      $print ENGINE1.DAT/que=la210_1

   c. From the ENGINE1.DAT determine:
      1) Pressure out of HPC
      2) Temperature out of the HPC
      3) Gamma at the HPC
      4) Fuel-air ratio (f) from the burner

   d. Run WRCOMP1A using these values as inputs and make a guess for RREF.

   e. If min doesn't equal mout from WRCOMP, then change the guess for RREF and iterate until they are equal.

   f. Run ENGINE1A in the wave rotor configuration with the correct value for RREF.

B. SAMPLE RESULTS

```
FILE WRPERF1.OUT FROM WRCOMP1A.FOR PROGRAM:

    # OF TIME STEPS K=          2000
    CFLNUM=    0.6000000
    # OF RIEMANN ITER. (QSTOP)=  20
    GRID CELL WIDTH DX= 0.01
    PTO    =    79.20000     psi
    PTOTIN=    546064.8      Pa
    RTIN    =    0.1956737      lbm/ft^3
    RTOTIN =      3.134375      kg/m^3
    PSEXIT1 =      75.54148     psi
    PSEXIT  =      520840.2     Pa
    PREF1 =     188.8537       (psi)
    PREF  =     1302101.       (Pa)
    xref  =    0.1800000       m
    xref1 =    0.5905512       ft

TIMING AND MASS FLOW BALANCE:
    INLET PORT OPENS AT:
    N= 741      TTOTAL=      1.2576689  TIME=      0.0003768
    EXHAUST PORT CLOSES AT:
    N=1014      TTOTAL=      1.7417805  TIME=      0.0005218
    INLET PORT CLOSES AT:
    N=1681      TTOTAL=      3.1865280  TIME=      0.0009547
    P(199)=0.685905  P(201)=0.684240  P(203)=0.684240
    Min= 0.561596  Mout= 0.626005      SWL=   1  SWR=   1

 WAVE ROTOR PERFORMANCE MAP ANALYSIS:
    SPR=  0.40  TPR  =    1.60  TTR  =1.1356  FIN =0.0165  AMEXIT=0.9000
    PTO= 79.20  TOTIN= 1056.00  GAMMA=1.3710  RTIN=0.1957  RTOTIN=3.1344
    PTE=126.72  PE   =   75.54  PREF =188.85  TTE = 1964.0 TE     = 1707.4
   TREF= 2187.9 RGUESS=0.22520  RREF= 3.6073

COMPUTER RUN TIME =    835.5391     secs.
```

C. WRCOMP PROGRAM LISTING

Sections which are commented out with a "C" contain features that have been disabled (eg. DISSPLA) or they contain amplifying remarks.

```
C       PROGRAM WRCOMP WITH VAN DER CORPUT SAMPLING AND SINGLE TIME STEP
        INTEGER QPRINT, QSTOP,SWL,SWR
        REAL MASSIN,MASOUT
        DIMENSION WNORM(12),IDIGT(12),XARRAY(100)
        DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
        COMMON/SUBS/P,R,U,A,S,X
        COMMON/GLIMM1/PGLIM,RGLIM,UGLIM,PL,RL,UL,PR,RR,UR,AL,AR,GL,GR.EPS
        COMMON/GLIMM2/DT,DX,XI
        COMMON/FUN1/G,PA,RA,UA,RB,RMU
        COMMON/SAMPLE/WNORM,IDIGT
        COMMON XARRAY,N1
C        CALL COMPRS
C        CALL BLOWUP(1.5)
C        CALL PAGE(11.0,8.5)
C        CALL HWSCAL('SCREEN')
C        DATA K,SWL,SWR/2000,1,2/
C        DATA N,CFLNUM,TTOTAL/0,0.60,0.0/
C        DATA PTOTIN,RTOTIN,PSEXIT/547292.900,3.250,497390.1/
C        DATA PREF,RREF,XREF/1243475.4,3.60,0.1800/
         DATA SWL,SWR/1,2/
C----------------------------------------------------------------
         OPEN(UNIT=20,FILE='WRPERF1.OUT',STATUS='NEW')
         WRITE(6,*)'INPUT NUMBER OF TIME STEPS (2000):'
         READ(5,*)K
         WRITE(20,*)'FILE WRPERF1.OUT FROM WRCOMP1A.FOR PROGRAM:'
         WRITE(20,*)
         WRITE(20,*)'     # OF TIME STEPS K= ',K
         WRITE(6,*)'INPUT CFL NUMBER: (RANGE 0.2 < CFLNUM <0.8)'
         READ(5,*)CFLNUM
         WRITE(20,*)'     CFLNUM= ',CFLNUM
C-------
         WRITE(6,*)'INPUT MAX NUMBER OF RIEMANN ITERATIONS (20):'
         READ(5,*)QSTOP
         WRITE(20,3021)QSTOP
 3021   FORMAT(5X,'# OF RIEMANN ITER. (QSTOP)= ',I3)
         WRITE(6,*)'INPUT GRID CELL WIDTH (0.01):'
         READ(5,*)DX
         WRITE(20,3022)DX
 3022   FORMAT(5X,'GRID CELL WIDTH DX= ',F4.2)
C-------INPUT TOTAL CONDITIONS INTO THE WAVE ROTOR--------------
         WRITE(6,*)'INPUT TOTAL PRESSURE IN (PTO) (psi):'
         READ(5,*)PTO
           PTOTIN=PTO*144.*47.88026
         WRITE(6,*)'     PTO    =',PTO,'psi'
         WRITE(6,*)'     PTOTIN=',PTOTIN,'Pa'
         WRITE(20,*)'     PTO    =',PTO,'psi'
         WRITE(20,*)'     PTOTIN=',PTOTIN,'Pa'
         WRITE(6,*)'INPUT TOTAL TEMP IN (TTOTIN) (R):'
         READ(5,*)TTOTIN
         WRITE(6,*)'INPUT GAMMA IN:'
         READ(5,*)GAMMA
           G=GAMMA
         WRITE(6,*)'INPUT FUEL-AIR RATIO IN (FIN):'
         READ(5,*)FIN
                 MGIN=(1.+FIN)/(0.034522+0.035648*FIN)
                 RGIN=1545.43/MGIN
c---------density in lbm/ft^3
           RTIN=PTO*144.0/(RGIN*TTOTIN)
C---------density in kg/m^3
           RTOTIN=RTIN*0.45359*(1./0.3048)**3.
```

```
      WRITE(6,*)'    RTIN   = ',RTIN,' lbm/ft^3'
      WRITE(6,*)'    RTOTIN = ',RTOTIN,' kg/m^3'
      WRITE(20,*)'    RTIN   = ',RTIN,' lbm/ft^3'
      WRITE(20,*)'    RTOTIN = ',RTOTIN,' kg/m^3'
      WRITE(6,*)'INPUT TOTAL PRESSURE RATIO (TPR):'
      READ(5,*)TPR
      WRITE(6,*)'INPUT EXIT MACH (ABSOLUTE):'
      READ(5,*)AMEXIT
C-------COMPUTE EXIT STATIC PRESSURE
       PSEXIT1=PTO*TPR/(1+(G-1.)/2.*AMEXIT**2.)**(G/(G-1.))
       PSEXIT=PSEXIT1*144.0*47.88026
      WRITE(6,*)'    PSEXIT1 = ',PSEXIT1,'psi'
      WRITE(6,*)'    PSEXIT  = ',PSEXIT,'Pa'
      WRITE(20,*)'    PSEXIT1 = ',PSEXIT1,'psi'
      WRITE(20,*)'    PSEXIT  = ',PSEXIT,'Pa'
C-------INPUT REF DATA-----------------------------------
      WRITE(6,*)'INPUT STATIC PRESSURE RATIO (SPR):'
      READ(5,*)SPR
C-------COMPUTE PREF------------------------------------
        PREF1=PSEXIT1/SPR
        PREF=PREF1*144.0*47.88026
      WRITE(6,*)'    PREF1 = ',PREF1,' (psi)'
      WRITE(6,*)'    PREF  = ',PREF,' (Pa)'
      WRITE(20,*)'    PREF1 = ',PREF1,' (psi)'
      WRITE(20,*)'    PREF  = ',PREF,' (Pa)'
      WRITE(6,*)'INPUT GUESS FOR RREF (lbm/ft^3):'
      READ(5,*)RGUESS
        RREF=RGUESS*0.45359*(1./0.3048)**3.
      WRITE(6,*)'    RGUESS = ',RGUESS,' lbm/ft^3'
      WRITE(6,*)'    RREF   = ',RREF,' kg/m^3'
C-------INPUT XREF IN METERS----------------------------
      WRITE(6,*)'INPUT XREF: (0.1800 m)'
      READ(5,*)XREF
C-------CONVERT XREF TO FEET FOR CORRECT DIMENSIONS ------
        XREF1=XREF/0.3048
      WRITE(6,*)'    xref  = ',XREF,' m'
      WRITE(6,*)'    xref1 = ',XREF1,' ft'
      WRITE(20,*)'    xref  = ',XREF,' m'
      WRITE(20,*)'    xref1 = ',XREF1,' ft'
C-----------------------------------------------------------------
C     G=1.35
C     GL=1.35
C     GR=1.35
      GL=GAMMA
      GR=GAMMA
      EPS=1.E-06
C     QSTOP=20
      N1=0
      JCOUNT=0
      KCOUNT=0
      UEXMAX=0.
      MASSIN=0.
      MASOUT=0.
C-----------------START COMPUTER TIME CLOCK--------------
      RINTT=0.0
      RCTIME=SECNDS(RINTT)
      OPEN(UNIT=15,FILE='WRCOMP1.DAT',STATUS='NEW')
C     DX=0.01
      AREF=SQRT(PREF/RREF)
      TIMREF=XREF/AREF
```

```
        RMU=SQRT((G-1.)/(G+1.))
        X(1)=-0.5*DX
        ZETA=WDP(1)
        XII=DX*(WDP(0)-0.5)
      DO 25 I=2,203
        X(I)=X(I-1)+0.5*DX
   25 CONTINUE
      DO 35 I=1,100
        XARRAY(I)=X(I*2+1)
   35 CONTINUE
C     INITIAL DATA
C     CALL INIT1
C     CALL INIT2L(PSEXIT,PREF,RREF)
      CALL INIT2R(PSEXIT,PREF,RREF)
C     CALL INIT3L(PSINL,RINL)
C     CALL INIT3R(PSINR,RINR)
C     NONDIMENSIONALIZATION
      DO 30 I=1,203,2
        P(I)=P(I)/PREF
        R(I)=R(I)/RREF
        U(I)=U(I)/AREF
        A(I)=A(I)/AREF
        S(I)=ALOG(P(I)/R(I)**G)
   30 CONTINUE
C      CALL PLOT1(K)
      DO 40 J=1,K
        N=N+1
        XII=DX*(WDP(0)-0.5)
        QPRINT=N/50
        DT=100.
        DO 50 I=1,203,2
          DTT=CFLNUM*DX/(2.*AMAX1(ABS(A(I)+U(I)),ABS(A(I)-U(I))))
          DT=AMIN1(DTT,DT)
   50   CONTINUE
        TTOTAL=TTOTAL+DT
        TIME=TTOTAL*TIMREF
        XI=-XII
        DO 60 I=1,201,2
          PL=P(I)
          RL=R(I)
          UL=U(I)
          PR=P(I+2)
          RR=R(I+2)
          UR=U(I+2)
          XITEMP=XI
          IF(I.EQ.1) XI=ABS(XI)
          IF((I.EQ.201).AND.(XI.GT.0.0)) XI=-XI
          CALL GLIMM(QSTOP,PSTAR,USTAR,ASTAR)
          XI=XITEMP
          P(I+1)=PGLIM
          R(I+1)=RGLIM
          U(I+1)=UGLIM
   60   CONTINUE
      CALL GG(SWL,SWR,N,TTOTAL,TIME,UEXMAX,PTOTIN,PREF,MASSIN,MASOUT)
        DO 70 I=1,201,2
          IF(XI.LT.0.) GOTO 80
          P(I+2)=P(I+1)
C        WRITE(6,*)'P(',I+2,')=',P(I+2)
          R(I+2)=R(I+1)
C        WRITE(6,*)'R(',I+2,')=',R(I+2)
```

```
          U(I+2)=U(I+1)
C        WRITE(6,*)'U(',I+2,')=',U(I+2)
          A(I+2)=SQRT(G*P(I+2)/R(I+2))
C        WRITE(6,*)'A(',I+2,')=',A(I+2)
          S(I+2)=ALOG(P(I+2)/R(I+2)**G)
C        WRITE(6,*)'S(',I+2,')=',S(I+2)
          GOTO 70
   80     P(I)=P(I+1)
          R(I)=R(I+1)
          U(I)=U(I+1)
          A(I)=SQRT(G*P(I)/R(I))
          S(I)=ALOG(P(I)/R(I)**G)
   70   CONTINUE
        IF(SWL.EQ.1) CALL BCL1
        IF(SWL.EQ.2) CALL BCL2(PSEXIT,PREF)
        IF(SWL.EQ.3) CALL BCL3(PSINL,RINL,PREF,RREF)
        IF(SWL.EQ.4) CALL BCL4(PTOTIN,RTOTIN,PREF,RREF)
        IF(SWL.EQ.5) CALL BCL5
        IF(SWR.EQ.1) CALL BCR1
        IF(SWR.EQ.2) CALL BCR2(PSEXIT,PREF)
        IF(SWR.EQ.3) CALL BCR3(PSINR,RINR,PREF,RREF)
        IF(SWR.EQ.4) CALL BCR4(PTOTIN,RTOTIN,PREF,RREF)
        IF(SWR.EQ.5) CALL BCR5
        IF(SWL.EQ.4) MASSIN=MASSIN+R(3)*U(3)*DT
        IF(SWR.EQ.2) MASOUT=MASOUT+R(201)*U(201)*DT
C       IF(N.EQ.(QPRINT*50)) CALL PLOT2(N,K)
   40 CONTINUE
C      CALL ENDPL(0)
C      CALL DONEPL
C
C --- --CALCULATE PRESSURES AND TEMPS FROM INPUT INFO--------
        PTE=PTO*TPR
        PE=PTE/((1.+(G-1.)/2.*AMEXIT**2.)**(G/(G-1.)))
        PREF=PE/SPR
        TREF=PREF*144./(RGIN*RGUESS)
        TE=TREF*(SPR**((G-1.)/G))
        TTE=TE*(1.+(G-1.)/2.*AMEXIT**2.)
C------CALC TOTAL TEMP RATIO FROM REF DENSITY GUESS---------
        TTR1=TPR**((G-1.)/G)
C
        WRITE(20,*)
        WRITE(20,3024)
        WRITE(20,3025)SPR,TPR,TTR1,FIN,AMEXIT
        WRITE(20,3026)PTO,TTOTIN,G,RTIN,RTOTIN
        WRITE(20,3027)PTE,PE,PREF,TTE,TE
        WRITE(20,3028)TREF,RGUESS,RREF
 3024  FORMAT(2X,'WAVE ROTOR PERFORMANCE MAP ANALYSIS:')
 3025  FORMAT(5X,'SPR=',F6.2,2X,'TPR  =',F8.2,2X,'TTR  =',F6.4,2X,
     >'FIN =',F6.4,2X,'AMEXIT=',F6.4)
 3026  FORMAT(5X,'PTO=',F6.2,2X,'TOTIN=',F8.2,2X,'GAMMA=',F6.4,2X,
     >'RTIN=',F6.4,2X,'RTOTIN=',F6.4)
 3027  FORMAT(5X,'PTE=',F6.2,2X,'PE   =',F8.2,2X,'PREF =',F6.2,2X,
     >'TTE =',F7.1,1X,'TE    =',F7.1)
 3028  FORMAT(4X,'TREF=',F7.1,1X,'RGUESS=',F7.5,2X,'RREF= ',F6.4)
        RCTIME=SECNDS(RINTT)-RCTIME
        WRITE(6,*)'COMPUTER RUN TIME = ',RCTIME,' secs.'
        WRITE(15,*)'COMPUTER RUN TIME = ',RCTIME,' secs.'
        WRITE(20,*)
        WRITE(20,*)'COMPUTER RUN TIME = ',RCTIME,' secs.'
        CLOSE(UNIT=15)
```

```
        CLOSE(UNIT=20)
      STOP
      END
C***********************************************************************
      SUBROUTINE GLIMM(QSTOP,PSTAR,USTAR,ASTAR)
      INTEGER Q,QSTOP
      REAL ML,MR,MLN,MRN
      COMMON/GLIMM1/PGLIM,RGLIM,UGLIM,PL,RL,UL,PR,RR,UR,AL,AR,GL,GR,EPS
      COMMON/GLIMM2/DT,DX,XI
      DATA Q,ML,MR/0,100.,100./
      PSTAR=0.5*(PL+PR)
      COEFL=SQRT(PL*RL)
      COEFR=SQRT(PR*RR)
      ALPHA=1.
C     BEGIN GODUNOV ITERATION
   30 Q=Q+1
      IF(PSTAR.LT.EPS) PSTAR=EPS
C     COMPUTE NEXT ITERATION FOR ML AND MR
      MLN=COEFL*PHI(PSTAR,PL)
      MRN=COEFR*PHI(PSTAR,PR)
      DIFML=ABS(MLN-ML)
      DIFMR=ABS(MRN-MR)
      ML=MLN
      MR=MRN
C     COMPUTE NEW PSTAR
      PTIL=PSTAR
      PSTAR=(UL-UR+PL/ML+PR/MR)/(1./ML+1./MR)
      PSTAR=ALPHA*PSTAR+(1.-ALPHA)*PTIL
      IF(Q.LE.QSTOP) GOTO 10
      IF(ABS(PSTAR-PTIL).LT.EPS) GOTO 20
C     COMPUTE NEW ALPHA
      ALPHA=0.5*ALPHA
      Q=0
      IF((1.-ALPHA).LT.EPS) GOTO 20
   10 IF(DIFML.GE.EPS) GOTO 30
      IF(DIFMR.GE.EPS) GOTO 30
C     END OF GODUNOV ITERATION; COMPUTE USTAR
   20 USTAR=(PL-PR+ML*UL+MR*UR)/(ML+MR)
C     BEGIN SAMPLING PROCEDURE
      IF (XI.LT.USTAR*DT) GO TO 40
C     RIGHT SIDE; SELECT CASE OF SHOCK OR EXPANSION
      IF (PSTAR.LT.PR) GO TO 50
C     RIGHT WAVE IS A SHOCK WAVE
      WR=UR+MR/RR
      IF (XI.LT.WR*DT) GO TO 60
C     RIGHT OF RIGHT SHOCK CASE
      RGLIM=RR
      PGLIM=PR
      UGLIM=UR
      RETURN
C     LEFT OF RIGHT SHOCK CASE
   60 RGLIM=-MR/(USTAR-WR)
      PGLIM=PSTAR
      UGLIM=USTAR
      RETURN
C     RIGHT WAVE IS A RAREFACTION WAVE
   50 CONST=PR/RR**GR
      RSTAR=(PSTAR/CONST)**(1./GR)
      ASTAR=SQRT(GR*PSTAR/RSTAR)
      AR=SQRT(GR*PR/RR)
```

```
      IF (XI.GE.(USTAR+ASTAR)*DT) GO TO 70
C     LEFT OF RIGHT FAN CASE
      RGLIM=RSTAR
      UGLIM=USTAR
      PGLIM=PSTAR
      LTAG=3
      RETURN
C     SELECT RIGHT OF FAN OR IN FAN
   70 IF (XI.GE.(UR+AR)*DT) GO TO 80
C     IN RIGHT FAN CASE
      UGLIM=2./(GR+1.)*(XI/DT-AR+(GR-1.)/2.*UR)
      RGLIM=(((AR+(GR-1.)/2.*(UGLIM-UR))**2.)/(GR*CONST))**(1./(GR-1.))
      PGLIM=CONST*RGLIM**GR
      RETURN
C     RIGHT OF RIGHT FAN CASE
   80 RGLIM=RR
      PGLIM=PR
      UGLIM=UR
      RETURN
C     LEFT SIDE; SELECT CASE OF SHOCK OR RAREFACTION
   40 IF (PSTAR.LT.PL) GO TO 90
C     LEFT WAVE IS A SHOCK WAVE
      WL=UL-ML/RL
      IF (XI.GE.WL*DT) GO TO 100
C     LEFT OF LEFT SHOCK CASE
      RGLIM=RL
      PGLIM=PL
      UGLIM=UL
      RETURN
C     RIGHT OF LEFT SHOCK CASE
  100 RGLIM=ML/(USTAR-WL)
      PGLIM=PSTAR
      UGLIM=USTAR
      RETURN
C     LEFT WAVE IS A RAREFACTION WAVE
   90 CONST=PL/RL**GL
      RSTAR=(PSTAR/CONST)**(1./GL)
      ASTAR=SQRT(GL*PSTAR/RSTAR)
      AL=SQRT(GL*PL/RL)
      IF (XI.LT.(USTAR-ASTAR)*DT) GO TO 110
C     RIGHT OF LEFT FAN CASE
      RGLIM=RSTAR
      PGLIM=PSTAR
      UGLIM=USTAR
      RETURN
C     SELECT LEFT OF FAN OR IN FAN CASE
  110 IF (XI.LT.(UL-AL)*DT) GO TO 120
C     IN LEFT FAN CASE
      UGLIM=2./(GL+1.)*(AL+(GL-1.)/2.*UL+XI/DT)
      RGLIM=(((AL+(GL-1.)/2.*(UL-UGLIM))**2.)/(GL*CONST))**(1./(GL-1.))
      PGLIM=CONST*RGLIM**GL
      RETURN
C     LEFT OF LEFT FAN CASE
  120 RGLIM=RL
      PGLIM=PL
      UGLIM=UL
      RETURN
      END
C******************************************************************
      FUNCTION PHI(Y,Z)
```

```
      REAL RMU
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      EPS=1.E-06
      PARAM=Y/Z
      IF (ABS(1.-PARAM).GE.EPS) GO TO 10
      PHI=SQRT(G)
      RETURN
   10 IF (PARAM.GE.1.) GO TO 20
      PHI=(G-1.)/2.*(1.-PARAM)/(SQRT(G)*(1.-PARAM**((G-1.)/(2.*G))))
      RETURN
   20 PHI=SQRT((G+1.)/2.*PARAM+(G-1.)/2.)
      RETURN
      END
      FUNCTION PHI1(PB)
      REAL RMU
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      PHI1=(PB-PA)*SQRT((1.-RMU**2.)/(RA*(PB+RMU**2.*PA)))
      RETURN
      END
C**********************************************************************
      FUNCTION PSI(PB)
      REAL RMU
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      PSI=SQRT(1.-RMU**4.)/RMU**2./SQRT(RA)*PA**(1./(2.*G))*(PB**((G-1.)
     >/(2.*G))-PA**((G-1.)/(2.*G)))
      RETURN
      END
C***** ****************************************************************
      SUBROUTINE INIT1
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      COMMON/SUBS/P,R,U,A,S,X
      DO 10 I=1,9,2
      P(I)=810600.00
      R(I)=0.7132
      U(I)=644.4
      A(I)=SQRT(G*P(I)/R(I))
   10 CONTINUE
      DO 20 I=11,203,2
      P(I)=101325.0
      R(I)=1.22
      U(I)=0.0
      A(I)=SQRT(G*P(I)/R(I))
   20 CONTINUE
      RETURN
      END
C*************************************************************
      SUBROUTINE INIT2R(PSEXIT,PREF,RREF)
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      COMMON/SUBS/P,R,U,A,S,X
      DO 10 I=3,201,2
      P(I)=PREF
      R(I)=RREF
      U(I)=0.0
      A(I)=SQRT(P(I)*G/R(I))
   10 CONTINUE
      P(1)=P(3)
      R(1)=R(3)
      U(1)=-U(3)
```

```
      A(1)=SQRT(G*P(1)/R(1))
      P(203)=PSEXIT
      R(203)=R(201)
      PA=P(201)
      RA=R(201)
      UA=U(201)
      PB=P(203)
      RB=R(203)
      IF(PA.GT.PB) GO TO 20
      U(203)=UA-PHI1(PB)
      GO TO 30
   20 U(203)=UA-PSI(PB)
   30 A(203)=SQRT(G*P(203)/R(203))
      RETURN
      END
C*****************************************************************
      SUBROUTINE INIT2L(PSEXIT)
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      COMMON/SUBS/P,R,U,A,S,X
      DO 10 I=3,201,2
      P(I)=285080.0
      R(I)=0.897
      U(I)=0.0
      A(I)=SQRT(G*P(I)/R(I))
   10 CONTINUE
      RETURN
      END
C************************************************************
      SUBROUTINE INIT3L(PSINL,RINL)
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      COMMON/SUBS/P,R,U,A,S,X
      DO 10 I=3,201,2
      P(I)=2390000.0
      R(I)=9.787
      U(I)=0.0
      A(I)=SQRT(G*P(I)/R(I))
   10 CONTINUE
      P(1)=PSINL
      R(1)=RINL
      PA=P(3)
      RA=R(3)
      UA=U(3)
      PB=P(1)
      U(1)=UA+PHI1(PB)
      A(1)=SQRT(G*P(1)/R(1))
      P(203)=P(201)
      R(203)=R(201)
      U(203)=-U(201)
      A(203)=SQRT(G*P(203)/R(203))
      RETURN
      END
C**************************************************************
      SUBROUTINE INIT3R(PSINR,RINR)
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      DO 10 I=3,201,2
      P(I)=2421667.5
```

```
      R(I)=9.787
      U(I)=0.0
      A(I)=SQRT(G*P(I)/R(I))
   10 CONTINUE
      P(203)=PSINR
      PA=P(201)
      RA=R(201)
      UA=U(201)
      PB=P(203)
      U(203)=UA-PHI1(PB)
      R(203)=RINR
      A(203)=SQRT(G*P(203)/R(203))
      P(1)=P(3)
      R(1)=R(3)
      U(1)=-U(3)
      A(1)=SQRT(G*P(1)/R(1))
      RETURN
      END
C*****************************************************************
      SUBROUTINE BCL1
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      P(1)=P(3)
      R(1)=R(3)
      U(1)=-U(3)
      A(1)=SQRT(G*P(1)/R(1))
      RETURN
      END
C*****
      SUBROUTINE BCR3(PSINR,RINR,PREF,RREF)
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      P(203)=PSINR/PREF
      R(203)=RINR/RREF
      PA=P(201)
      RA=R(201)
      UA=U(201)
      PB=P(203)
      U(203)=UA-PHI1(PB)
      A(203)=SQRT(G*P(203)/R(203))
      RETURN
      END
C*****
      SUBROUTINE BCL3(PSINL,RINL,PREF,RREF)
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      P(1)=PSINL/PREF
      R(1)=RINL/RREF
      PA=P(3)
      RA=R(3)
      UA=U(3)
      PB=P(1)
      U(1)=UA+PHI1(PB)
      A(1)=SQRT(G*P(1)/R(1))
      RETURN
      END
C*****
```

```
      SUBROUTINE BCR1
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      P(203)=P(201)
      R(203)=R(201)
      U(203)=-U(201)
      A(203)=SQRT(G*P(203)/R(203))
      RETURN
      END
C*****
      SUBROUTINE BCL2(PSEXIT,PREF)
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      P(1)=PSEXIT/PREF
      R(1)=R(3)
      U(1)=U(3)
      A(1)=SQRT(G*P(1)/R(1))
      RETURN
      END
C*****
      SUBROUTINE BCR2(PSEXIT,PREF)
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      P(203)=PSEXIT/PREF
      R(203)=R(201)
      PA=P(201)
      RA=R(201)
      UA=U(201)
      PB=P(203)
      RB=R(203)
      IF(PA.GT.PB) GO TO 10
      U(203)=UA-PHI1(PB)
      GO TO 20
   10 U(203)=UA-PSI(PB)
   20 A(203)=SQRT(G*P(203)/R(203))
      RETURN
      END
C*****************************************************
      FUNCTION WDP(II)
      DIMENSION WNORM(12),IDIGT(12)
      COMMON/SAMPLE/WNORM,IDIGT
      IF (II.EQ.0) GO TO 10
      L1=2
      L2=1
      DO 20 JJ=1,12
      IDIGT(JJ)=0
      WNORM(JJ)=1./FLOAT(L1**JJ)
   20 CONTINUE
      WDP=0.
      RETURN
   10 DO 40 JJ=1,12
      L1=2
      L2=1
      KJ0=IDIGT(JJ)
      KJN=MOD((KJ0+1),L1)
      IDIGT(JJ)=KJN
      IF (KJ0.LT.KJN) GO TO 50
```

```
   40 CONTINUE
   50 SUM=0.
      DO 60 JJ=1,12
      KNEW=MOD(IDIGT(JJ)*L2,L1)
      SUM=SUM+FLOAT(KNEW)*WNORM(JJ)
   60 CONTINUE
      WDP=SUM
      RETURN
      END
C*********************************************************************
C       SUBROUTINE PLOT2(N,K)
C       DIMENSION XORG(4),YORG(4),YMAX(4),YMIN(4),KNT(4),IYNAM(10)
C       DIMENSION PARRAY(100),RARRAY(100),UARRAY(100),SARRAY(100),XARRAY(1
C      >00)
C       DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
C       COMMON/SUBS/P,R,U,A,S,X
C       COMMON XARRAY
C       DATA XORG/0.5,4.75,0.5,4.75/
C       DATA YORG/0.5,0.5,4.75,4.75/
C       DATA YMAX/2.50,2.5,2.5,2.5/
C       DATA YMIN/0.0,0.0,-1.0,-2.0/
C       DATA KNT/1,4,6,9/
C       DATA IYNAM/'PRES','SURE','$   ','DENS','ITY$','VELO','CITY','$   '
C      >,'ENTR','OPY$'/
C       DO 200 I=1,100
C       PARRAY(I)=P(I*2+1)
C       RARRAY(I)=R(I*2+1)
C       UARRAY(I)=U(I*2+1)
C       SARRAY(I)=S(I*2+1)
C   200 CONTINUE
C       DO 300 I=1,4
C       CALL PHYSOR(XORG(I),YORG(I))
C       CALL AREA2D(3.5,3.5)
C       CALL XNAME('X',1)
C       CALL YNAME(IYNAM(KNT(I)),100)
C       CALL GRAF(0.,'SCALE',1.0,YMIN(I),'SCALE',YMAX(I))
C       IF(I.EQ.1) CALL SETCLR('YELLOW')
C       IF(I.EQ.2) CALL SETCLR('CYAN')
C       IF(I.EQ.3) CALL SETCLR('RED')
C       IF(I.EQ.4) CALL SETCLR('MAGENTA')
C       IF(N.EQ.K) CALL SETCLR('WHITE')
CC      IF(N.EQ.2398) CALL SETCLR('BLUE')
C       IF(I.EQ.1) CALL CURVE (XARRAY,PARRAY,100,0)
C       IF(I.EQ.2) CALL CURVE (XARRAY,RARRAY,100,0)
C       IF(I.EQ.3) CALL CURVE (XARRAY,UARRAY,100,0)
C       IF(I.EQ.4) CALL CURVE (XARRAY,SARRAY,100,0)
C       CALL ENDGR(0)
C   300 CONTINUE
C       RETURN
C       END
C*********************************************************
C       SUBROUTINE PLOT1(K)
C       DIMENSION XORG(4),YORG(4),YMAX(4),YMIN(4)
C       DATA XORG/0.5,4.75,0.5,4.75/
C       DATA YORG/0.5,0.5,4.75,4.75/
C       DATA YMAX/2.50,2.5,2.5,2.5/
C       DATA YMIN/0.0,0.0,-1.0,-2.0/
C       DO 10 I=1,4
C       CALL PHYSOR(XORG(I),YORG(I))
C       CALL AREA2D(3.5,3.5)
```

```
C      CALL FRAME
C      CALL GRAF(0.,'SCALE',1.0,YMIN(I),'SCALE',YMAX(I))
C      CALL ENDGR(0)
C   10 CONTINUE
C      CALL PHYSOR(8.5,0.5)
C      CALL AREA2D(2.25,7.75)
C      CALL FRAME
C      CALL GRAF(0.,'SCALE',1.,0,'SCALE',K)
C      CALL ENDGR(0)
C      RETURN
C      END
C****************************************************************
      SUBROUTINE BCL4 (PTOTIN,RTOTIN,PREF,RREF)
      INTEGER QOUT
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      DIMENSION QR(3)
      DIMENSION XARRAY(100)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      COMMON XARRAY,N1
      N1=N1+1
      QR(3)=2./(G-1.)*A(3)-U(3)
      QOUT=N1/5
      PTOT=PTOTIN/PREF
      RTOT=RTOTIN/RREF
      ATOT=SQRT(G*PTOT/RTOT)
      STOT=ALOG(PTOT/RTOT**G)
      U(1)=U(3)
      A(1)=SQRT(ATOT**2.-(G-1.)/2.*ABS(U(1))**2.)
      QR(1)=2./(G-1.)*A(1)-U(1)
      AMACH=U(1)/A(1)
      IF(AMACH.LT.0.0) GO TO 60
      P(1)=PTOT/(1.+(G-1.)/2.*AMACH**2.)**(G/(G-1.))
      R(1)=RTOT/(1.+(G-1.)/2.*AMACH**2.)**(1./(G-1.))
      S(1)=ALOG(P(1)/R(1)**G)
      GO TO 50
   60 P(1)=PTOT/(1.+(G-1.)/2.*ABS(AMACH)**2.)**(G/(G-1.))
      R(1)=RTOT/(1.+(G-1.)/2.*ABS(AMACH)**2.)**(1./(G-1.))
      S(1)=ALOG(P(1)/R(1)**G)
   50 IF(N1.EQ.5*QOUT) WRITE(10,1) A(1),A(3),U(1),U(3),N1
    1 FORMAT(2X,4F9.6,5X,I5)
      RETURN
      END
C*****
      SUBROUTINE BCR4 (PTOTIN,RTOTIN,PREF,RREF)
      INTEGER QOUT
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      DIMENSION QL(203)
      DIMENSION XARRAY(100)
      COMMON/SUBS/P,R,U,A,S,X
      COMMON/FUN1/G,PA,RA,UA,RB,RMU
      COMMON XARRAY,N1
      N1=N1+1
      QL(201)=2./(G-1.)*A(201)+U(201)
      QOUT=N1/25
      PTOT=PTOTIN/PREF
      RTOT=RTOTIN/RREF
      ATOT=SQRT(G*PTOT/RTOT)
      STOT=ALOG(PTOT/RTOT**G)
      U(203)=U(201)
```

```
      A(203)=SQRT(ATOT**2.-(G-1.)/2.*ABS(U(203))**2.)
      QL(203)=2./(G-1.)*A(203)+U(203)
      AMACH=U(203)/A(203)
      IF(AMACH.LT.0.0) GO TO 60
      P(203)=PTOT/(1.+(G-1.)/2.*AMACH**2.)**(G/(G-1.))
      R(203)=RTOT/(1.+(G-1.)/2.*AMACH**2.)**(1./(G-1.))
      S(203)=ALOG(P(203)/R(203)**G)
      GO TO 50
   60 P(203)=PTOT/(1.+(G-1.)/2.*ABS(AMACH)**2.)**(G/(G-1.))
      R(203)=RTOT/(1.+(G-1.)/2.*ABS(AMACH)**2.)**(1./(G-1.))
      S(203)=ALOG(P(203)/R(203)**G)
   50 IF(N1.EQ.25*QOUT) WRITE(10,1) P(201),U(201),N1,N
    1 FORMAT(2X,2F13.6,5X,2I5)
      RETURN
      END
C*****
      SUBROUTINE BCL5
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      P(1)=P(3)
      R(1)=R(3)
      U(1)=U(3)
      A(1)=A(3)
      RETURN
      END
C*****
      SUBROUTINE BCR5
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      P(203)=P(201)
      R(203)=R(201)
      U(203)=U(201)
      A(203)=A(201)
      RETURN
      END
C***********************************************************************
      SUBROUTINE CC(SWL,SWR,N,TTOTAL,TIME,UEXMAX,PTOTIN,PREF,MASSIN,MASO
     *UT)
      INTEGER SWL,SWR
      REAL MASSIN,MASOUT
      DIMENSION P(203),R(203),U(203),A(203),S(203),X(203)
      COMMON/SUBS/P,R,U,A,S,X
      IF((SWL.EQ.1).AND.(SWR.EQ.2)) GO TO 10
      IF((SWL.EQ.4).AND.(SWR.EQ.2)) GO TO 30
      IF((SWL.EQ.4).AND.(SWR.EQ.1)) GO TO 50
      IF((SWL.EQ.1).AND.(SWR.EQ.1)) RETURN
   10 PWALL=P(2)
      IF(PWALL.LE.(PTOTIN/PREF)) GO TO 20
      RETURN
   20 SWL=4
      WRITE(6,74)
      WRITE(6,75) N,TTOTAL,TIME
       WRITE(15,74)
       WRITE(15,75) N,TTOTAL,TIME
        WRITE(20,*)
        WRITE(20,*)'TIMING AND MASS FLOW BALANCE:'
       WRITE(20,74)
       WRITE(20,75) N,TTOTAL,TIME
      RETURN
   30 UEXIT=U(202)
```

```
      IF(UEXMAX.LT.UEXIT) UEXMAX=UEXIT
      IF(UEXIT.LT.UEXMAX/2.) GO TO 40
      RETURN
   40 SWR=1
      WRITE(6,76)
      WRITE(6,75) N,TTOTAL,TIME
       WRITE(15,76)
       WRITE(15,75) N,TTOTAL,TIME
       WRITE(20,76)
       WRITE(20,75) N,TTOTAL,TIME
      RETURN
   50 P1SHOK=P(2)
C     IF(N.EQ.1590) GO TO 60
      IF(P1SHOK.GT.PTOTIN/PREF) GO TO 60
      RETURN
   60 SWL=1
      WRITE(6,77)
        WRITE(15,77)
        WRITE(20,77)
      WRITE(6,75) N,TTOTAL,TIME
        WRITE(15,75) N,TTOTAL,TIME
        WRITE(20,75) N,TTOTAL,TIME
      WRITE(6,78) P(199),P(201),P(203)
        WRITE(15,78) P(199),P(201),P(203)
        WRITE(20,78) P(199),P(201),P(203)
      WRITE(6,73) MASSIN,MASOUT,SWL,SWR
        WRITE(15,73) MASSIN,MASOUT,SWL,SWR
        WRITE(20,73) MASSIN,MASOUT,SWL,SWR
   73 FORMAT(5X,'Min=',F9.6,2X,'Mout=',F9.6,5X,'SWL=',I4,2X,'SWR=',I4)
   74 FORMAT(5X,'INLET PORT OPENS AT:')
   75 FORMAT(5X,'N=',I4,5X,'TTOTAL=',F14.7,2X,'TIME=',F14.7)
   78 FORMAT(5X,'P(199)=',F8.6,2X,'P(201)=',F8.6,2X,'P(203)=',F8.6)
   76 FORMAT(5X,'EXHAUST PORT CLOSES AT:')
   77 FORMAT(5X,'INLET PORT CLOSES AT:')
      RETURN
      END
```

APPENDIX C

COMPUTER SYSTEM AIDS

A. TRANSFERRING FILES

1. FROM MAINFRAME TO AERO PC LAB

The following procedures use SIMPC to download programs from the NPS mainframe to the Aero Department's PC Lab. The modem and SIMPC hooked up to the IBM PC-XT-2 in the Aero PC lab. The following **boldfaced** commands outline the procedures.

```
C\ <dir> >cd\SIMPC.........change dir to SIMPC dir
C\SIMPC>SIMPC ACCESS.SIM...execute SIMPC macro ACCESS.SIM
...
...
LOGON: 9812P     ....logon to mainframe (user number)
PASSWORD: _____           ....enter your mainframe password
...
...
...
<enter>      ....executes your PROFILE.EXEC
...
...
FORSIMPC  GET  ENGINE  FORTRAN  A1..file  to  copy  from
                  mainframe
...
...
A:ENGINE.FOR
...
...
FORSIMPC PRINT ENGINE FORTRAN A1..optional if you wanted
              to send your file to the current printer.
LOGOFF       ...logoff the mainframe
<ALT> <G>    ...hangs up the modem
<ALT> <Q>    ...quits SIMPC and returns to DOS
```

The procedure above uses the macro ACCESS.SIM which was written specifically to link between the Aero PC lab nd the NPS mainframe.

2. FROM AERO P.C. LAB TO VAX SYSTEM

The following procedure uses PROCOMM, an executable modem link, and ALT 0, a modem macro written specifically to link the Aero PC lab to the Aero VAX lab. KERMIT is the actual file transferring program.

```
C\ <dir> >cd\COM...change to COM directory.
PROCOMM .executable program
<ALT> 0 .macro modem link
ATDT2953
CONNECT
USERNAME: GUEST    .gouge username
PASSWORD: _____    .password will not appear

Are you logged on from a PC? Y
Do you want to connect to WASP? Y

Username: ______    .typical VAX logon
Password: ______    .

C\>KERMIT   ..executes file transfer program

to send a file:
Receive  RCOMP1.FOR...Program you wish to call
F3: WR  P.FOR  ... file you wish to send but this file
    must be in the PROCOMM dir on the C (hard) drive!

LOGOFF      ...logoff the VAX
<ALT> x ...exits PROCOMM
```

If you have any difficulties F10 is the help menu!

B. USING "SIGHT" UTILITY ON THE VAX

SIGHT is a utility drawing program available on the Aero Department VAX computer. It was used to create pop-up menus for the configurations for the ENGINE program. To run SIGHT, at the DCL prompt enter

```
$SIGHT
```

A pop-up menu provides simple instructions. Save your drawing as a <fn>.UIS file and then use the VAX "render" command to create your output file as a <fn>.REN file. An example for sending creating and sending an output file to the laser (ln03) printer is:

```
$RENDER <fn>.UIS/DEVICE_TYPE=ln03
$PRINT/PASSALL/QUE=ln03_fn.REN
```

SIGHT can be used to generate a picture from inside a program. This can be done by inserting the following statement in a (FORTRAN) program.

```
CALL LIB$SPAWN ('RENDER <fn>.UIS')
```

This was done for the ENGINE program.

C. GRAFkit PROCEDURES

PLOT3.FOR is a GRAFkit plotting program which was incorporated in the ENGINE1A.FOR program. PLOT3.FOR will be available as an engine performance plotting program on the VAX network. It can accept up to 50 separate data inputs for a selected X-axis, a Y1-axis for specific thrust, and a Y2-axis for specific fuel consumption. The instructions for running PLOT3.FOR are included in the comments section of the program listing.

PLOT3.FOR program listing

```
C-----THIS PLOTTING PROGRAM USES GRAFKIT SOFTWARE---------
C-----ENSURE THE FOLLOWING SEQUENCE IS ENTERED PRIOR TO
C     EXECUTION OF PLOT3:
C       logon the VAX
C         $GKSETUP
C         $UIS
C         $RUN PLOT3
C-----IF YOU WANT YOUR RESULTS SENT TO EITHER THE LASER OR
C       HIGH SPEED PRINTERS:
C       logon the VAX
C         $GKSETUP
C         $ln03
C         $DEFINE GK_OUT PLOT.DAT       ("PLOT.DAT" IS YOUR)
C         $ln03s_ep                     (OUTPUT fn.ft      )
C         $RUN PLOT3
C....... ..to send output to the selected printer:
C         $PRINT/PASSALL/QUE=LN03 PLOT.DAT
C-------------------------------------------------------
      PROGRAM PLOT3
      REAL X1(50),Y1(50),Y2(50)
      CHARACTER X1LABEL*10,Y1LABEL*33,Y2LABEL*33
      CHARACTER FIG1*7,NUMB1*3,TOPLBL*10
C========================================================
C-----SPECIFY X,Y LEFT AND Y RIGHT LABELS------------------
C
        Y1LABEL='A: Specific Thrust    lbf/(lbm/s)'
        Y2LABEL='B: SFC   (lbm fuel/hr)/lbf thrust'
C-----CHOOSE THE X PARAMETER TO PLOT
      WRITE(6,*)'CHOOSE YOUR X AXIS LABEL(i.e. 1):'
        WRITE(6,*)'    1. BYPASS RATIO'
        WRITE(6,*)'    2. FAN PRESSURE RATIO'
        WRITE(6,*)'    3. COMPRESSOR RATIO'
        WRITE(6,*)'    4. LPT EXIT MACH NUMBER'
        WRITE(6,*)'    5. OTHER (MAX 10 CHARACTERS)'
         READ(5,*)IANS4
        IF (IANS4.EQ.1) THEN
          X1LABEL=' BETA '
         ELSEIF (IANS4.EQ.2) THEN
          X1LABEL=' PRLC '
         ELSEIF (IANS4.EQ.3) THEN
          X1LABEL=' PRHC '
         ELSEIF (IANS4.EQ.4) THEN
          X1LABEL='AMACH8'
         ELSEIF (IANS4.EQ.5) THEN
          WRITE(6,*)' INPUT X AXIS LABEL:'
          READ(5,'(A10)')X1LABEL
        ENDIF
```

```
C
C-----CHOOSE THE NUMBER OF DATA PTS TO PLOT MAX=50-----------
C
      WRITE(6,*)'INPUT NUMBER OF DATA POINTS TO PLOT (MAX=50)'
        READ(5,*) ICP1
C-----INPUT X AXIS VALUES:
C
      WRITE(6,*)'INPUT VALUES FOR ',X1LABEL
        DO I=1,ICP1
          WRITE(6,1000)'X(',I,') ='
          READ(5,*)X1(I)
        END DO
C-----


        WRITE(6,*)'INPUT: '
        WRITE(6,*)'   0 - AUTOMATICALLY DETERMINE X1MIN AND X1MAX'
        WRITE(6,*)'   1 - INPUT X1MIN AND X1MAX'
         READ(5,*)IANS5
          IF (IANS5.EQ.1) THEN
             WRITE(6,*)' INPUT X1MIN =:'
             READ(5,*)X1MIN
             WRITE(6,*)' INPUT X1MAX =:'
             READ(5,*)X1MAX
C-----       AUTOMATICALLY DETERMINE THE MIN AND MAX X VALUE
          ELSE
             X1MIN=X1(1)
             X1MAX=X1(1)
             DO J=1,ICP1
               X1TEMP=X1(J)
               X2TEMP=X1(J)
               IF (X1MIN.GT.X1TEMP) THEN
                 X1MIN=X1TEMP
               ENDIF
               IF (X1MAX.LT.X2TEMP) THEN
                 X1MAX=X2TEMP
               ENDIF
              END DO
           ENDIF
C-----INPUT VALUES FOR Y1 AND Y2 ARRAYS
      WRITE(6,*)'INPUT VALUES FOR ',Y1LABEL
        DO J=1,ICP1
          WRITE(6,1000)'Y1(',J,') ='
          READ(5,*)Y1(J)
        END DO
      WRITE(6,*)'INPUT VALUES FOR ',Y2LABEL
        DO K=1,ICP1
          WRITE(6,1000)'Y2(',K,') ='
          READ(5,*)Y2(K)
        END DO
 1000 FORMAT(1X,A3,I2,A3)
C
C-----INPUT THE NUMBER OF THE FIGURE TO PLOT (FIG 1)---
        WRITE(6,*)'INPUT THE FIGURE NUMBER (ie 1 OR 2...)'
          READ(5,'(A9)')NUMB1
C       WRITE(6,*)'NUMB1=',NUMB1
        FIG1='FIGURE '
        TOPLBL=FIG1//NUMB1
C       WRITE(6,'(1X,A11)')'TOP LABEL =',TOPLBL
C-----------------------------------------------------------
C-----CONSTRUCT THE PLOT
C
      WRITE(6,*)'STANDBY FOR PLOT'
C-----OPEN GKS, OPEN AND ACTIVATE WORKSTATION.
        CALL OPNGKS(1,2,1)
C
C     SET UP THE COLOR TABLE
        CALL GSCR(1,0,0.00,0.00,0.50)
        CALL GSCR(1,1,1.00,1.00,0.00)
        CALL GSCR(1,2,0.00,1.00,0.50)
        CALL GSCR(1,3,0.00,1.00,1.00)
        CALL GSCR(1,4,1.00,0.00,0.00)
        CALL GSCR(1,5,1.00,0.00,1.00)
        CALL GSCR(1,6,1.00,0.00,1.00)
        CALL GSCR(1,7,1.00,1.00,1.00)
```

```
      CALL GSCR(1,8,0.00,0.00,0.00)
C
C-----SET VIEWPOINT AND DEFINE THE RANGE OF USER'S DATA TO BE USED
      CALL AGSETF ('GRID/LEFT.',    .15)
      CALL AGSETF ('GRID/RIGHT.',   .85)
      CALL AGSETF ('GRID/TOP.',     .85)
      CALL AGSETF ('GRID/BOTTOM.',  .15)
C-----SET UP FOR FIRST PLOT
C----- DEFINE THE TOP, BOTTOM, LEFT AND RIGHT LABELS
C----- PUT BLANKS IN FIRST CURVE LABELS
      CALL AGSETC ('LABEL/NAME.',  'T')
      CALL AGSETI ('LINE/NUMBER.', 100)
      CALL AGSETC ('LINE/TEXT.',   ' ')
C
      CALL AGSETC ('LABEL/NAME.',  'B')
      CALL AGSETI ('LINE/NUMBER.',-100)
      CALL AGSETC ('LINE/TEXT.',   ' ')
C
      CALL AGSETC ('LABEL/NAME.',  'L')
      CALL AGSETI ('LINE/NUMBER.', 100)
      CALL AGSETC ('LINE/TEXT.',   ' ')

      CALL AGSETC ('LABEL/NAME.',  'R')
      CALL AGSETI ('LINE/NUMBER.',-100)
      CALL AGSETC ('LINE/TEXT.',   ' ')
C-----TURN OFF RIGHT AXIS
      CALL AGSETF ('AXIS/RIGHT/CONTROL.',0.0)
C-----SET YMIN AND YMAX FOR FIRST PLOT
      CALL AGSETF ('X/MINIMUM.', X1MIN)
      CALL AGSETF ('X/MAXIMUM.', X1MAX)
      CALL AGSETF ('Y/MINIMUM.',   0.0)
      CALL AGSETF ('Y/MAXIMUM.', 120.0)
C------
      CALL AGSTUP (X1,1,1,ICP1,1,Y1,1,1,ICP1,1)
      CALL AGBACK
      CALL GSPLCI (7)
      CALL DASHDC ('$$$$$$$$$$$''''''''''$$$$$$$$$$',3,1)
C------PLOT CURVE 1: X1 VS Y1
      CALL AGCURV (X1,1,Y1,1,ICP1,-1)
C
C-----------------------
C-----SET UP FOR SECOND PLOT
C----- DEFINE THE TOP, BOTTOM, LEFT AND RIGHT LABELS
C----- THE LARGER LINE NUMBER WRITES ABOVE PREVIOUS LABELS
      CALL AGSETC ('LABEL/NAME.',  'T')
      CALL AGSETI ('LINE/NUMBER.', 100)
      CALL AGSETC ('LINE/TEXT.',TOPLBL)
C
      CALL AGSETC ('LABEL/NAME.',  'B')
      CALL AGSETI ('LINE/NUMBER.',-110)
      CALL AGSETC ('LINE/TEXT.',X1LABEL)
C
      CALL AGSETC ('LABEL/NAME.',  'L')
      CALL AGSETI ('LINE/NUMBER.', 110)
      CALL AGSETC ('LINE/TEXT.',Y1LABEL)

      CALL AGSETC ('LABEL/NAME.',  'R')
      CALL AGSETI ('LINE/NUMBER.', -100)
      CALL AGSETC ('LINE/TEXT.',Y2LABEL)
C     TURN OFF THE TOP, BOTTOM AND LEFT AXES
```

```
      CALL AGSETF ('AXIS/TOP/CONTROL.',0.0)
      CALL AGSETF ('AXIS/BOTTOM/CONTROL.',0.0)
      CALL AGSETF ('AXIS/LEFT/CONTROL.',0.0)
C-----TURN ON THE RIGHT AXIS
      CALL AGSETF ('AXIS/RIGHT/CONTROL.',1.0)
C-----CHANGE THE YMIN AND YMAX
      CALL AGSETF ('Y/MINIMUM.', 0.0)
      CALL AGSETF ('Y/MAXIMUM.', 1.5)
C-----TURN ON THE RIGHT AXIS NUMERICS
      CALL AGSETF ('RIGHT/NUMERIC/TYPE.',1.E6)
C
      CALL AGSTUP (X1,1,1,ICP1,1,Y2,1,1,ICP1,1)
      CALL AGBACK
      CALL GSPLCI (4)
      CALL DASHDC ('$$SFC$''''',8,1)
C------PLOT CURVE 2: X1 VS Y2
      CALL AGCURV (X1,1,Y2,1,ICP1,-2)
C-----CALL FRAME TO ADVANCE THE FRAME
C-----CLOSE GKS, CLOSE AND DEACTIVATE WORKSTATION
      CALL FRAME
      CALL CLSGKS
C------------------------
      END
```

LIST OF REFERENCES

1. NPS-67-85-008, Proceedings of the 1985 ONR/NAVAIR/ Wave Rotor Research and Technology Workshop, May 1985.

2. Taussig, Robert T., "Wave Rotor Turbofan Engines for Aircraft," Mechanical Engineering, V. 60, November 1984.

3. NPS-67-86-008CR, Estimation of a Turbofan Engine Cycle Performance with and without a Wave Rotor Component, by Atul Mathur, September 1986.

4. NPS-67-86-006CR, Code Development for Turbofan Engine Cycle Performance with and without a Wave Rotor Component, by Atul Mathur, September 1986.

5. NPS-57Va73061A, Calculating Procedure of Sea-Level Static Performance of Two-Spool Afterburning Bypass Jet Engine, by M.H. Vavra, June 1973.

6. NASA TM X-1340, Computer Program for Design-Point Performance of Turbojet and Turbofan Engine Cycles, by Michael R. Vanco, February 1967.

7. Mattingly, J., Heiser, W.H., Daley, D.H., Aircraft Engine Design, American Institute of Aeronautics and Astronautics, Inc., 1987.

8. Oates, Gordon C., Aerothermodynamics of Gas Turbine and Rocket Propulsion, American Institute of Aeronautics and Astronautics, Inc., 1988.

9. NPS-67-85-006CR, Wave Rotor Research: A Computer Code for Preliminary Design for Wave Diagrams, by A. Mathur, May 1985.

10. Glimm, James, "Solutions in the Large for Nonlinear Hyperbolic Systems of Equations," Communication on Pure and Applied Mathematics, V. xviii, pp. 697-715, 1965.

11. Chorin, Alexandre Joel, "Random Choice Solution of Hyperbolic Systems," Journal of Computational Physics, V. 22, pp. 517-533, 1976.

12. Sod, Gary A., "A Survey of Several Finite Difference Methods for Systems of Nonlinear Conservation Laws," Journal of Computational Physics, V. 27, pp. 1-31.

13. Anderson, John D., Jr., Introduction to Flight, McGraw-Hill Book Co., pp. 569-572, 1989.

14. GRAFkit Reference Manual, version 3.1-2, Vols. 1 and 2, SCO Inc., 1989.

15. Kentfield, J.A.C. and O'Blenes, M., "Methods for Achieving a Combustion-Driven Pressure Gain in Gas Turbines," Transactions of the ASME, V. 110, pp.704-711, October 1988.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Department Chairman, Code AA Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, California 93943-5000	1
4.	Professor Raymond P. Shreeve, Code AA/Sf Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, California 93943-5000	6
5.	LT James W. Roberts 1440 Mulberry Lane Lemoore, California 93245	2
6.	Naval Air Systems Command (Attn: G. Derderian AIR 536) Washington, D.C. 20361	1
7.	Professor Garth V. Hobson, Code AA/Hg Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, California 93943-5000	1